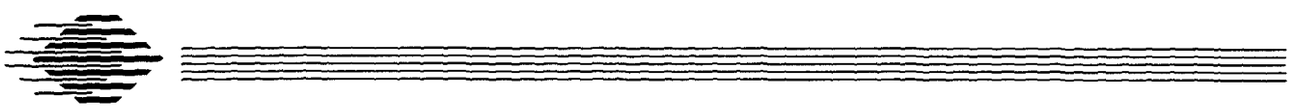
Technical Report
CMU/SEI-96-TR-013
ESC-TR-96-013
July 1, 1996

Software Process Automation: Experiences from the Trenches



Alan Christie
Linda Levine
Edwin J. Morris
David Zubrow
Software Engineering Institute

Teresa Belton
Larry Proctor
Nolan Norton And Company

Denis Cordelle
Jean-Eloi Ferotin
Jean-Philippe Solvay
Cap Gemini Segoti

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

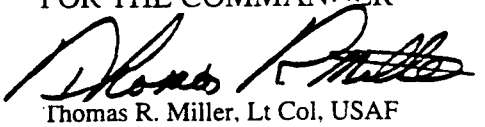
Revised copy

This report was prepared for the

SEI Joint Program Office
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Table of Contents

List of Figures

Software Process Automation: Experiences from the Trenches

Abstract: Software process automation is a new technology with significant promise. However practical experience in the field is still limited and there appears to be a variety of potential barriers to its use. The objective of this empirical study is to document current practical experience and to identify what works and what does not. Lessons learned from the study will be disseminated to help others who wish to implement the technology. This report documents results from the first phase of the study in which 14 in-depth interviews were conducted. Personnel interviewed were involved in projects in which process-centered environments were developed and adopted.

1 Purpose of the Study

Awareness of software process and its improvement has increased significantly over the past decade. This impact can be measured by the number of organizations that are improving their process capability through Capability Maturity Model [Paulk 93] of the SEI, the rapidly expanding interest in software engineering process groups (SEPGs), and the development of ISO standards focused on software, etc. [ISO 91]. However, the application of technology to support these process activities has been remarkably absent. Technology can automate process elements such as collecting metrics, ensuring process repeatability, reusing processes, guiding novices through processes, and supporting training.[1] However, this support is not yet common. The goal of this study is to examine the reasons why technology has not been more frequently adopted as a contributor to software process effectiveness. Thus we wanted to get answers to such questions as

- Are process automation products still too immature to be effective?
- Are poor adoption strategies hindering introduction?
- Is process automation too confining on end users?
- Are software processes too complex to be supported by automation?
- Are there simply insufficient examples of successful use of the technology?

Our goal is not only to help end-user organizations, but also those who are developing tools with which process-centered environments can be built: that is, researchers and commercial tool vendors. Thus the specific objectives of the study are to

1. In this study, we focus primarily on computer support for processes in which activity sequences and/or artifact state changes are important elements in guiding process flow. Other types of computer support for product development such as groupware, are not discussed. See Appendix A for a more detailed review of terms relevant to this study.

1. Identify the technical, social, and organizational inhibitors to the adoption of process automation:

   - Assess the prevalence and scope of software process automation.
   - Categorize the technologies and practices that are currently being used.
   - Identify effective and ineffective technologies and practices.
   - Develop guidelines for process automation implementers.

2. Support vendors and researchers in developing products more in tune with end-user needs:

   - Evaluate current research and vendor technologies in light of experiences gained in 1) above.
   - Develop guidelines for researchers and vendors to improve product effectiveness.
   - Foster effective communications between researchers, vendors, developers and end users.

These objectives are being met through a series of activities that include in-depth interviews followed by a questionnaire survey and a workshop. The objectives of these activities are as follows:

- The interviews are aimed at gathering practitioner experiences in a relatively unstructured way, to identify what individuals believe are the important issues in the adoption of software process automation, and to establish a basis for the more structured questionnaire survey. Results from these interviews are the main focus of this report.
- The questionnaire survey, to be discussed in a future report, will assess a wider cross-section of those involved with process automation, and will include individuals outside the software community. Because the questionnaire respondents are following a standard format, the data in this phase of the study will be analyzed in a more quantitative fashion. It is anticipated that respondents to the questionnaire will be contacted about a year after the survey. This will allow us to estimate what progress (or lack thereof) organizations have made over an extended period of time, and to identify why some projects have been successful and why some have failed.
- Finally, the workshop is aimed at identifying success strategies for the introduction of software automation. It is intended that the workshop bring together a widely diverse group of individuals with experience in research and development, adoption, management and end use of process automation, and to raise awareness of critical issues across these communities.

Three independent organizations are involved in performing the study: the SEI, Nolan Norton and Company (a division of KPMG Peat Marwick), and Cap Gemini Sogeti (located in Grenoble, France).

Section 2 of the report provides a brief overview of who we interviewed and how we analyzed the information obtained from the interviewees. Section 3 briefly summarizes the characteristics of the interviewed organizations. Section 4 then organizes the information we heard into a set of findings categories, using extensive quotes to support the described findings.

Because of the lack of standard terminology in the field, Appendix A defines a set of process-related terms used throughout the report, Appendix B provides a copy of the interview script that supported all the interviews, and Appendix C provides summaries of all fourteen interviews.

2 How We Approached the Interviews

This report is based upon interviews of individuals who were knowledgeable about and experienced with process automation. We performed a qualitative analysis of these interviews to arrive at the findings reported here.

2.1 Interviewees

An extensive list of candidates was identified early on, including end-user organizations, commercial and in-house developers, and researchers. Our original goal was to interview mostly end users of process automation. However, that was not to be. Because of the immaturity of the technology, we interacted with relatively few experienced end users of the technology. Most of our interviews were with people who were involved in developing and implementing process-centered environments (PCE)s.

These individuals came from a wide variety of organizations including

- a vendor of a major process-oriented configuration management (CM) product,
- four DoD sites implementing process-centered environments (PCEs),
- two US government contractors who were developing process tools and implementing PCEs,
- two French government contractors who were implementing PCEs,
- a French bank that is operating with a PCE,
- a university group with strong ties to industry.

2.2 The Interviews

A total of 14 interviews were conducted with 12 projects.[1] In the large majority of these interview sessions, two interviewers were present. The number of interviewees in each interview ranged from one to eight. All interviews were taped to ensure that the comments were recorded accurately. The interviews took approximately 36 hours with an average length of 2.4 hours per interview. All in all, the interviews yielded 150 pages of transcripts.

A standard script supported each interview. The script identified hypotheses to be examined, which were followed by questions pertaining to the issues in which we were interested. This script provided a consistent framework and ensured that we would have comparable information from each of the interviews. While the questions were used to support the interviews, and

1. In one organization, two different projects were interviewed. With two other projects, multiple interviews were conducted.

to assure coverage, they were not followed mechanically: Often areas of interest were probed in depth. The topics covered (and the related hypotheses) are as follows:

- Business/Product characteristics: Business/product environment defines the types of processes and hence the effectiveness of automated processes.
- Process maturity: One must be a process-mature organization to use process automation effectively.
- Application focus: Some types of processes are more appropriate for automation than others.
- Use of tools and technical development: A familiarity with technology prior to the automation project is a prerequisite to successful automation.
- Team characteristics and experiences: A range of both technical and non-technical competencies is required to implement process automation technology.
- Transition, adoption, and management: A well thought-out transition/ adoption strategy is critical to end-user acceptance.
- Impacts and insights: Successful use of process automation should correlate with the six areas identified above.

The interview script is provided in Appendix B.

2.3 Analysis of Results

The analysis of the interview results was informal and consisted of two activities: email discussion of issues identified in the interviews and meetings to brainstorm and summarize key findings. The email discussion was conducted in the following manner: Interviewers initially identified issues and wrote up potential findings based on the issues. Other interviewers then commented on these proposed findings based upon the observations from the interviews they conducted. In all, 24 email messages covering 8 themes (e.g., technocentric approach to process automation) were composed.

This dialog served as input to a team meeting in which all interviewers met to further discuss the findings. In particular, we sought to reach consensus on the findings and to organize them around a small number of topics.

3 Overview of the Projects

Table 3-1 is a summary of the projects studied, including the classification of the project, such as technology demonstration (demo), experiment (exp), or full scale deployment (fsd). In addition, we identify whether the project is a commercial or military activity, and provide its current status. Table 3-2 summarizes the major tools, technologies employed, and significant issues that were identified by the interviewees.

These tables are provided to introduce the reader to the individual projects. We will refer to these projects further during the course of the discussion.

Table 3-1: Application Characteristics of Projects

Project	Activity	Duration	Size of Organiz.	Type of Organization	Lifecycle Component
A	Developing a process centered environment intended for general use	Jan 91 - Present	60-80	Military	Maintenance
B	Developing process tool	5 years		Commercial/ gov. funded	Maintenance
C	Developing of a PCE, intended for commercial sale	1 year		Commercial	Project scheduling
D	Command/Control, Inactive	Oct 92 +3 years		Military	
E	Creating simplified (less process-centered) version of the tool	5 years?		Commercial tool vendor	Full
F	Experimental Research	Ongoing	1 prof.+ students	Academic	NA
G	MIS, Inactive (abandoned)	2.5 years	10-15	Military	Cleanroom, Maintenance
H	Inactive, but some effort to find commercialization interest	5 years	10 - 100 potential	Commercial	2167A S/W development
I	Automated problem tracking	Ongoing	5-10	Commercial	Maintenance
J	Automating specification/quality control and development/validation	Experimental/ongoing	1-5	Commercial	Development/ Maintenance
K	Automating a problem tracking/-maintenance scenario	Pilot Effort	1-5	Commercial	Maintenance
L	Automating a reverse engineering Scenario	Ongoing, Pilot	10-15	Military	Maintenance

Table 3-2: Technology Characteristics of Projects

Proj.	Technology	Supporting Tools	Major Issues
A	Synervision	CM, Language parsing (reengineering), static/dynamic analysis, document preparation, project management, requirements traceability	Process definition, selective use, support
B	Process Weaver, FlowMark, building own	Stp, Interleaf, Paradigm Plus, Oracle	Money unavailable to buy licenses, Not-invented-here syndrome
C	Process Weaver	Schedule Publisher, Oracle, Interleaf, Worldview, OpenInterface	
D	Process Weaver, and Custom process front ends	CAT/Compass, Amadeus, and contractor-developed software products	Resistance to massive amount of technology, Integration of technologies, conflicting points of view between adopting org and consultants
E	CM	Frame	Labor/resource intensive, time consuming adoption, complex tool demands significant effort for adoption
F	System Factory Project	Internally developed tools to support modeling, analysis, simulation, visualization, enactment	
G	Process Weaver	ProcessWeaver, Oracle,	Tool Instability, design restrictions placed on end users
H	CASE	Atherton S/W backplane	Development time exceeded sponsorship and customer patience, expectation drift.
I	Process Weaver	Database (supporting problems and solutions)	Integration of problem database
J	Process Weaver	WordPerfect, All-in-One, Oracle, CM System	Integration of tools
K	Process Weaver	Framemaker, CM System	Integration of CM tool
L	InConcert	Cadre, AutoPlan, DBStar	Ineffective process integration, poor training, time-consuming environment maintenance

4 Interview Findings

The interviewees represented one or more automation efforts that, loosely speaking, can be seen as pilot projects. These projects ranged in size from fewer than 10 to more than 60 people. For purposes of discussion, the numbers cited include the personnel for whom the automation was intended, as well as the developers of the automation if they are part of the same organization. Typical project size was toward the low end.

While we made no attempt to formally measure the level of process maturity of the organizations/projects interviewed, some had previously undergone formal process assessments using the SEI CMM [Paulk 93]. These projects ranged in maturity from Level 1 (ad hoc/chaotic) to Level 5 (optimizing). However, most can be characterized as relatively immature (at or below Level 2). Other projects had not been formally assessed, but many characterized themselves as having a poorly defined set of software development processes. Two projects were attempting software development activities for the first time.

Of the twelve projects interviewed (seven currently active, four inactive, one experimental), only two were far enough along for the automation to be considered institutionalized. In one case, the automation was associated with a company that developed and distributed a configuration management product. This product has significant process capability that is used to support further development of the product. The other organization that effectively adopted PCE technology did so to support software problem tracking.

Four points may be made about the interviews and the findings derived from them. First, because of the immaturity of the technology, we interviewed few people who could be considered to be experienced end users of the technology. The great majority of interviewees were either developers of process-centered environments, developers of the process tools from which PCEs can be built, or managers of development projects. Second, the findings not only surfaced problems but identified potential solutions to these problems. We hope that this information will be useful to organizations intending to build and use PCEs. Third, interviewees' experiences were not always consistent, and these inconsistencies may at times be reflected in the report. Fourth, as might be expected, we found that many of the adoption issues we identified have much in common with adoption issues associated with other technology areas.

We divide the findings into three major categories

- drivers and inhibitors
- contributors to success
- technology issues

In the following discussions, we make heavy use of quotes (indicated in italics) from the interviews. A major reason for this is that interviewees were surprisingly frank in giving us their views about process automation and how their organizations were dealing with it.

4.1 Drivers and Inhibitors

4.1.1 Drivers

A variety of drivers was identified. In summary, these were

- cost reduction
- quality improvement
- maintaining process capability
- training
- project management support

We found that the most common issues driving organizations to process automation were the needs to remain competitive, to improve quality, and reduce costs. As a manager with a large government contractor stated: *The ideal is to do it cheaper, faster and improve quality. That's the reality of today's budgets.* In another government-related organization we heard: *The organization is currently downsizing, primarily by attrition. As personnel are transferred out, they are not replaced.* In an era of shrinking government budgets and increasing commercial competitive pressures, it is not surprising that solutions such as process automation are being explored.

Another automation project was strongly motivated by cost reductions. It estimated that the return on investment would be 17 times the initial investment over a period of 10 years, with the break-even point occurring in the first year. Numbers were also generated to suggest that without automation, 48 personnel would be needed, with automation half that number would be adequate. Given the current state of the project, these estimates are likely to be highly unrealistic, but they did convince management that automation was the way to go.

Another driver that we identified was related to the issue of maintaining process capability, particularly in organizations that were experiencing high turnover. In one DoD organization we heard: *Process is critical to organizations because there is such a high turnover of personnel. I have been on this program for two and one half years and almost everyone is new. There is a going away party every week or two.* Another manager indicated: *Management is pushing the sharing of information because you do not know how long you are going to be here... Younger people are saying "I have skills that can get me more money outside, so they are leaving."* Thus organizations are finding that they cannot afford to maintain processes in people's heads, even if there is adequate documentation and training.

Training was identified as an area that process automation could support. Guidance provided by the automated environment was seen as a benefit. This one interviewee stated: *[T]raining/ education will be greatly reduced with online context-sensitive help in leading you down the process. Instead of taking two years to get up to speed, it may take six months to get someone producing code.*

Project managers may be motivated to view process automation as a means to take control of the activities in the project. One interviewee suggested that senior management would like to see automated processes being schedule-driven. Speaking in this role, he indicated: *Any time you get a new work item it will key on a template that says... a change requires you to do these things and that will initiate lower level activities. This will allow me to track how many errors are being generated.*

Other expected drivers, such as support for process improvement or the automated collection of metrics, did not appear to be such significant factors. One example was described in which a drive was made to move a brand new project very quickly to CMM level 3. A process tool was introduced in an attempt to fix the resulting chaos. This simply compounded the problem.

4.1.2 Inhibitors

A number of inhibitors were identified. In summary these were

- reluctance to use someone else's technology
- lack of acceptance of external consultants
- resistance from "old hands"
- fears of first-line supervisors
- disjoint between predicted and actual times for implementation
- inability to achieve consensus on process definition
- inability to predict return on investment

These are discussed in more detail below.

A major inhibitor uncovered was the reluctance of people in an organization to accept process automation if it is perceived as being driven from outside that organization. As one interviewee stated: *Organizations are real resistant to change if it is perceived as being driven from outside the organization. "You developed that? then I won't use it." They may think they have a better idea and attempt to implement it. Then management edicts it and they use it as minimally as possible to be compliant. "I'll use it but use the minimal number of mouse clicks."* In one case inter-group resentment pitted groups in the organization against each other. The issue that arose was the perception by some groups that those chosen to be the test group for automation were not pulling their weight. The interviewee characterized the attitude with mock resentment: *These people have extra time on their hands, and we're over here dying!*

We heard from two consultants who independently voiced their frustration when working with clients. In one case, the consultant stated: *We put all our energies into developing the environment and went down and gave it to them. And they said "that's nice but we don't want it." That was an eye-opener.* In the other case, the consultant, who joined the project after it had started, said: *In my opinion, a good consultant will never come in and say "everything you are doing is bad." You can't say that, so what you have to do is back into these things.... The only way you can reach people is through education. I have to write things out for you and make*

them so painfully obvious that if you ignore them you get what you deserve. If I haven't done that then I have not done my job as a consultant.

In several organizations, "old hands" resisted the imposition of process automation, as they perceived this to be an intrusion into processes they knew well. This inhibitor was not present with less-experienced staff who more often welcomed the guidance that automation provided. One interviewee stated: *The new people were enthusiastic, but experienced analysts said "I hate the screens that tell me what I have to do -- can you make it so that there is a novice mode and an expert mode?"* Another interviewee stated: *Adoption is hard because some of the old hands are extremely resistant.*

The same interviewee also stated: *It's the first, second, and third line coordinators who really fear this stuff. In reality these are the people who should be contributing to this, because they really understand the process best.* Clearly, more senior people may resent the intrusion of process automation, as they have more to lose by its introduction. However, these may be the very people who can make the most valuable contributions to the design of the automated processes. Fear of being made redundant by automation may be very legitimate, and could be a significant inhibitor if not addressed appropriately.

Consistently we saw that it took people much longer than they anticipated to develop effective automated processes. One interviewee stated: *I think one reason why it's frustrating not getting product out is that it's taking much longer than everyone expects to go up the process automation learning curve.* Management expectations with respect to the investments required in process automation can be unrealistic. Thus we heard: *A reason for failure is the perception that process automation involves little more than the purchase of appropriate tools, and that tools are the major cost component. In reality, we found that adoption takes longer than everyone expects and that technical integration problems frequently cause unforeseen delays.* This was supported by one of the few cases where process automation was successfully institutionalized, and where technical issues outweighed organizational/people issues. In this case we heard: *The implementation has been quite long and difficult (8 months) but only for technical reasons.*

One of the activities for which schedules were significantly delayed was process definition; obtaining a consensus on what the detailed process should look like was often a long, drawn-out task. In one case, significant delays (i.e., years) were incurred: *While consensus can be reached at the higher levels of process, consensus on details was elusive. This appears to be due to differences in projects, differences in groups within a project, and differences in individuals.* The same interviewee indicated that: *The effort has involved two to six integrators/toolsmiths. A total of 19 person-years of effort was expended. However, the majority of the effort was in process definition. Most of the work had to be thrown away.* However, another interviewee identified the same problem and offered some insight: *The interesting thing is that the actual implementation step is not that difficult if you take a structured, engineering approach... What we've seen is many people spinning up front for years until they reach some definition of a process. The catch is: architecture first, then a phased process def-*

inition plan, and then do it a piece at a time. (This issue will be further discussed under "Using an Incremental Approach" in the next section.)

With respect to financial resources, one tool vendor we interviewed suggested that an inhibitor faced by management occurs because: *most companies have no way of figuring return-on-investment in their own organization. It is easy to identify up-front costs, but difficult to figure the ROI over a long period.*

4.2 Contributors to Success

A number of contributors that improve the chances for success were identified. These contributors were associated with

- obtaining management commitment
- addressing process definition issues
- encouraging communication
- providing training
- building effective development teams
- using an incremental approach
- minimizing risk

None of these issues is unique to process automation – they have to be faced when most complex technologies are introduced. However, because of the strong social impact of process automation, these issues are particularly challenging. Let us address each of these issues.

4.2.1 Obtaining Management Commitment

There is a need to sustain management commitment at all levels and throughout all phases of the automation project. Thus management expectations must be appropriately set. While different managers may react differently to process automation, we heard: *The people who were going to use the technology seemed to appreciate it – especially the system engineers. They are really tool-oriented. The first-line manager was against it while the second-line manager was for it.* This may reflect the fact that first-line managers have to take the impact of an unproven technology directly, potentially disrupting their schedules and commitments. Another interviewee indicated: *Make sure at the executive level that the expectations are set right, and that the limitations of the technology are understood. Many managers have the silver bullet syndrome – they all listen to the first good story they hear from a sales rep, and don't have anything to base their decision on other than the tool sounds good. Then it becomes law. That's how politics happens.*

Because process automation is still not a mature technology, convincing upper management to spend money may be challenging. We heard the opinion: *It's difficult to get management to spend money on something that they are not sure they see the value in. They have so many hot irons in the fire anyway... The main reason that we were successful was that we had a*

strong proponent over in the contractor office, and in the system area that was going to use it. He took a lot of initiative, maybe exceeding his authority in some cases.

4.2.2 Addressing Process Definition Issues

As mentioned previously, one of the most time consuming activities is defining processes that are acceptable to all interested parties. In one organization, the process was intended to support a wide range of individuals. This diversity made it extremely difficult to reach consensus. Initially, a detailed thirteen-step process was developed, but consensus could not be reached. Currently they have a seven-step, less-detailed process, and even with this more general process, obtaining consensus was difficult. Developers of the automated process felt that the primary problem was identifying requirements – as the processes changed over time, so did the requirements. Developers noted that even seemingly trivial changes to the process could have significant ripple effects on the system's implementation.

Consistent with that experience, an interviewee from another project stated: *Once the process is written down, review is a lot of trouble. I don't see any way around that. I don't that see improving the notation would help.*

Implementers may become carried away with the flexibility of a process technology and define processes that are too detailed. One interviewee stated: *Some of our customers get carried away with the flexibility of the tool to the point that they define very convoluted, sophisticated, complex processes, because they know what they can do with the tool. However, there is a start up cost associated with implementing that model. They use the model and find that they don't need all the bells and whistles they built in.*

One organization had developed a simple, low-tech approach to process definition that is worth repeating. Paraphrasing the interviewer's words: *We took their process step by step with free input. If I say the input is a frog, then nobody else challenges that. So I start with "what do you call the first process step?" Since they know their jobs, they all know what they do first and we put that activity's name at the top of the stack. And then I'll ask what triggers this and usually they'll perhaps say – it's some management directive. Generally I'll have two or three people writing "sticky notes"*[1] *because people are frequently throwing out ideas. Initially I was sticking them on to a large sheet of paper, but then Jose had the idea of arranging the sticky notes on the paper into a process sequence - while the others were thinking up ideas. As they were throwing out scenarios, I just tried to keep them from going too deep into subprocesses. After the process is defined, I set the paper aside and put up a new one for the next phase. Then they may say, "Oh, the exit criteria for the last phase are the entrance criteria for the next phase." From here on they usually get the hang of it. About one and a half hours is all that anybody can take of this at one stretch. On one project we pushed for about four hours and became brain-dead.*

1. 3M Post-It Notes™

On the issue of trying to extract process definitions from naive process users, there may be problems of completeness or enactability. One interviewee stated: *What they had was a process, but when we asked them to write it down, they did so in what they believed was a very detailed fashion. When you started to look at it, you would come to dead ends in the process. When you asked them about that, they'd say "well sometimes we do this and sometimes we do that." We told them when you put it into a computer you have to state this way or that, or flip a coin... Just the idea of having to code the process into a computer caused them to sit down and define the process to the level that the computer needed.*

4.2.3 Encouraging Communication

More than most software technologies, process automation requires close communication among those who are involved with it. This communication may be between technical staff and managers, or between members of different organizations. Mismatches in perception of what the technology will do for different roles may result in conflict. One interviewee stated: *The person leading the effort (one of the senior bean counters) said "here is my idea of a process architecture – it will be schedule-driven. Any time you get a new work item it will key on a template that says: a change [request] requires you to do these things, and that will initiate lower activities, and I will be able to track how many errors are being generated." People said this does not help me do my job. So his challenge at the end of the meeting was if you can come up with something better let me know, otherwise we are going to go forward and do this. That's when Bill and Mike came in from opposite ends of the spectrum and they went ahead and collaborated. Once we found out what real people needed to do their jobs, every bit of the data that the manager needed to view came from what they put in. When there are 240 worker bees to a dozen managers, I want the worker bees on my side. You show the managers that the metrics are going to be collected etc., you just need an SQL query to pull it out. Then the manager's light goes on.*

Another issue was isolation of the group developing the processes from those who will subsequently use the process. One interviewee stated: *The [end-user] group was reluctant because they were not included along the way. They perceived that processes were being developed off in a vacuum, then bestowed upon them.*

Voicing the need to involve all aspects of the organization that will be involved in process automation, another interviewee suggested: *You really need some technology advocates – people who can go proselytize out to the organization, people who are trusted in the organization. It's easy to get someone who wants to get on your band wagon, but who does not interface well with the group. So when you form your team get someone who is excited about it, and can go back and spread the word.*

4.2.4 Providing Training

Training developers in the technology and end users in the use of the automated processes is key to successful implementation. Because process automation technology does not pro-

duce a product (as does a compiler), it is harder to describe. One interviewee related the difficulty of describing process automation using a made-up dialog:

But what does [process automation] do?
It does your process.
Well, is it an editor?
No.
Is it a CASE tool?
No.
You mean I have to generate my own outputs?
Yes.
Then what advantage is it?
Well, it's a process tool.

In a similar vein, another interviewee stated: *There was a small group who understood [process automation's] value. There was a smaller group that even understood what it was trying to do. And a lot of people said "I just don't know what it is, but I don't even need it."*

These experiences indicate that end-user training needs to start with explaining the fundamental nature of process automation, and that training should not only focus on the detailed mechanics of what buttons to push and screens to fill in. Two other interviewees also suggested that simply holding training classes is not sufficient. One interviewee stated: *Training has been conducted for individual tools. Also the automation group spends much time doing hand holding, consulting in order to facilitate tool use.* The other interviewee said: *Training was provided in both tools and process. Initial training was provided four to five months before actually starting the job. Personnel had to be retrained and lots of hand-holding provided.* This last statement also suggests that timing of the training is critical to its effective use.

4.2.5 Building Effective Development Teams

Implementing process automation requires a development team with the correct mix of technical and organizational skills and a strong team leader. One interviewee saw that the credibility of a strong leader made a significant difference to acceptance: *He said "trust me, this will be good for you," and they believed him. This is not always going to happen, but this was a small tight team and it worked.* Another interviewee suggested that having a sufficiently senior person on the team was important: *You need a sufficiently senior person to capture the process to decide how deep or detailed you want to go. The tool can do anything you ask it to, but you do not want to have to excuse yourself to go to lunch.* However another interviewee was hesitant about having management on the process definition team: *Representatives come from across the organization, all different levels of people. In the development group, we have one of the senior coordinators, four developers from different areas of systems software, generic coding. Managers are not there – managers inhibit that kind of thing.*

Two other team-related items were heard. In the first, the interviewee indicated that *a special project room was set up to force project personnel to come together in one place and develop*

team spirit. Such structural changes are critical because you must break up the organization in order to get the necessary changes. Another interviewee suggested the following strategy: *The biggest advantage that we have and admittedly most companies don't is that the people we hire for development are people who are really into process, and want to do process automation.*

4.2.6 Using an Incremental Approach

The majority of people we interviewed indicated that their process automation strategy was one of the "great leap forward" variety. However most felt, in retrospect, that an incremental adoption approach should have been taken and that, given the state of the practice, the initial effort had been overly-ambitious. As described by one interviewee: *The baby steps approach says -- get them so far, get them acclimated, then bring in the new technology as they can appreciate it. If you try to bring an organization a big bag of technology, the first thing they will do is take the bag and put it in the garbage. So you have to bring in a piece at a time. It's got to be supportive of human activity and it's got to be very goal oriented, and produce immediate results.* With respect to an incremental approach, we also heard: *You need to bring [process automation] in a piece at a time. You need to see where they are at and how they are doing. Then pick one of their problems and try to solve that – so it's not too big. I think this is what we would do differently because we really didn't have a way to scale.*

A warning came from one interviewee with respect to management's expectations: *Management wants to see big bangs when they spend their money, not small steps. But the big bang approach doesn't work... They have to understand. Maybe software management education is needed to help them cross the chasm.* The same interviewee voiced the opinion: *If you are used to doing things in a certain way on your PC and they bring in a SUN Workstation with Interleaf and CADRE, it's too much. They can't change that fast. If you then tell them they are going to get a list, and you can only open the tools when the system tells you, people have a hard time with that.*

With respect to tool introduction, another interviewee suggested: *Process comes before [process] tools. We are very strong over that. A tool is a tool...You can't throw a switch and enact a process. Tools should be chosen to match your needs.* However, one interviewee indicated that having experience with application tools, prior to automating the process, made much sense: *Get [application] tools in use ASAP, even before automation is available. This gives users some experience, acceptance of the technology, as well as helping them define the real requirements.*

Finally, one interviewee suggested the following step-by-step adoption strategy: *I believe in starting on a pilot basis, defining a manually enactable process first. I'd be very reluctant to jump on [process] tools first. By manually implementing first, you wring out a whole lot of methodology issues and end up with good appreciation of what a balanced approach to the definition and enactment is. That will arm you with the ability to impose a set of quite realistic requirements on the next tool developer/ vendor who comes along and says I can solve your*

process problems. Talk to tool developers based on sound knowledge of what's really involved so that you will be less inclined to accept at face value what the tool developer says. Other thing - you don't have to swallow process automation all in one go. You can start with a database for metrics, defining artifacts and their states in repository - manage the artifacts, and let process drift by itself. Have people own and be responsible for changing artifacts from this state to that state by that date. Later add prescribed methods for doing these things, add process activities, link them together, define exit criteria, and form process network. By keeping process definition. divorced from management of artifacts, you get the flexibility to throw out a process that's not working well and substitute a new process without perturbing the products or artifacts that you are working on. You can add several processes, working from multiple viewpoints on the same artifact without perturbing the artifacts themselves. These things you learn from first enacting manually. Users may say "you didn't prioritize any of my activities but I wish you would. I have 30 activities I need help in prioritizing." But don't tell them what they have to do or it will be rejected. These things allow you to gradually work your way up the automation scale.

4.2.7 Minimizing Risk

Since experience with process automation is still limited, implementing a risk minimization strategy makes sense. Risk can come from many places and one project's risks may be quite different from another's. One tool vendor we interviewed was quite strong in suggesting that risk assessment should be part of the adoption effort, to be applied not only at the start of the automation project, but on a periodic basis throughout.The interviewee stated: *When I do risk management with the customer, out of it comes a set of risks. Generally I find that everyone gets about 30 risks. It always seems to work out to about 30. Even in-house for us, we came up with 30 risks when changing over to Lotus Notes. Only 10 percent to 25 percent came from the tool. Others are related to:*

- *what are the politics?*
- *what is the culture?*
- *what are the people issues?*
- *what are the legacy problems that people have never had the courage, or been able to solve?*

The interviewee suggested the following detailed risk categories:

- sponsorship
- network infrastructure
- resistance to change
- heterogeneous platform
- scalability issues
- training
- what processes are defined
- resources
- methodology
- tool integration
- legacy systems
- culture change
- tool limitations
- what processes to automate

- security
- network infrastructure
- system administration
- handling roll-out

While the interviewee suggested that serious risk can come from any category, in her estimation the first four (i.e., sponsorship, resources, network infrastructure, and methodology) often had the greatest impact.

4.3 Technology Issues

Process automation technology is still in its early days and interviewees (primarily PCE developers) suggested areas where capability could be improved. These areas are:

- end-user support
- tool/data integration
- technology support for process
- prototypes
- control driven versus artifact-state driven
- system performance

4.3.1 End-User Support

Several interviewees felt that the less imperative a process-centered environment was with the end user, the better. One interviewee stated: *The key is being unobtrusive. If you can do it and be unobtrusive then it is a win all around.* Supporting this sentiment was the view: *People, especially creative people, don't respond to a tool which says "you will start the activity now." For example, you cannot start implementing until your low-level design has been approved.* In other words, people feel more comfortable performing multiple tasks concurrently. Thus PCEs (and the underlying process-centered frameworks[1]) should provide mechanisms to allow this and should not place unnecessary restrictions on task sequencing.

There was a variety of other functional issues that were raised, and are simply quoted below:

On "to-do" lists: *We originally had the concept of a to do list. We would check off tasks and other tasks would appear. This is a very narrow view. Now we have the concept, not just of looking at today's to-do list but you can look at tomorrow, or next week, based on what we know now. It will be a best guess, based on durations and planning for future tasks.*

On interfacing with email and office scheduling: *An interface to email and a calendaring system would be very nice, because we like to keep online calendars to schedule meetings etc. You can go out and say "find me between this day and this day, a conference for a design review with these 5 people for an hour" and this could all be done automatically by the tool*

1. Process framework is used here to connote the product with which process-centered environments can be built. See Appendix A for a more detailed definition.

because it has all the information. Why should I pay a librarian to make a thousand phone calls?

On managing processes: *Every time someone had an instance of a process, they would take one of these forms, copy it, and put it in the book. So the process now becomes a book of forms. I said "This is silly - lets automate it and use that as a front end with which to manage the processes."*

4.3.2 Tool/Data Integration

Unfortunately tools often do not have consistent or compatible capabilities - a design tool may have overlapping functionality with a development tool, and each tool may present its information through a very different user interface. In addition, two tools that need to share data may use different data schema. These technical challenges can result in an integrated system that neither looks nor performs in an integrated manner. Unfortunately resolving functional, data, and user interface incompatibilities is usually a very hard problem. Another integration problem that surfaced in one organization was incompatibilities between two process definition notations that were used, and between these and the notations that were implemented in other process support tools.

Some of these problems were voiced by developers of PCEs, particularly the challenge of data integration between application tools embedded in the PCE. One interviewee stated: *The big data integration problem was between two tools. These had totally different views of process.* Another interviewee stated a similar opinion: *The technical problems can be worked but data integration is an exception - it's a hard problem.*

Tool integration concerns were described by a third interviewee: *[L]ike a lot of other things in the PCE, you find tools are not very well separated... As soon as you have a lot of different tools, all of which have their unique knowledge of process and artifact management, how do you get them to work together? If you take a total system view, there are encapsulation decisions you would ideally make if you were the PCE god, but which you can't do, because you are getting software off the shelf that has a lot of built-in assumptions.* The process management component provided primarily textual guidance on task activities while the application tools environment provided the technical means to carry out the task. In this way some of the integration issues were separated. As the interviewee stated: *we were not originally thinking this way. Then we saw that the application tools seems to cluster here while the process tools seems to cluster there. And there are some ties between them. Metrics are collected and displayed up here [through the process component] for management purposes. So there is a coupling through metrics.*

4.3.3 Technology Support for Process

A process-centered framework must provide a range of capabilities that help the PCE developer do his/her job. Areas where interviewees indicated that such support is desirable were:

- graphical modeling capability with which to build processes
- support for multiple role-perspectives
- a library of standardized process components that can be incorporated into larger processes

One interviewee felt strongly that graphical support was needed both in the process development and process execution phases. The next three quotes are from one interviewee who had experiences with both ProcessWeaver and FlowMark: With respect to process development he stated: *I think one of the great advantages to both of these tools is the graphical way you can build a process.*

He also noted the advantage of minimizing the time necessary to develop user interfaces: *ProcessWeaver has a feature that I really like. I don't need a GUI builder to build all the screens to interface – it builds its own forms, it has its own agendas, and work contexts, and that was very nice. In FlowMark, we had to go out and build our own screens, because it is only a process tool. If you want to put a panel in front of someone you have to build it yourself (there is no interface tool).*

This interviewee also suggested that both tools were harder to learn than he would have liked: *One of the things that people around here didn't like about both tools was that you had to be an expert. In other words, they would like a process tool which has a very simple English-like language.*

Another interviewee supported the need for a graphical development environment, in order to support process design reviews: *We felt we needed a tool we could use during process design. We needed to design a process, have someone come in and take a look at it, see if they approved of the way the tool interactions worked given, of course, that they recognized automation was coming.*

Finally one interviewee voiced the need for role-based views in the process: *If you are a developer you only see developer steps, not management or QA views, for example. The user just interacts through a To-Do list. It would come up and it would sort priority. The person could see it and select the task to do. The process manager would get the artifacts from the repository, check them out and return them to the user. It would also open up the tool.*

4.3.4 Prototypes

Divergent views were heard on the use of PCE prototypes. On the one hand we heard the view: *Prototypes are very formal around here. They are a big part of what we do. Fundamentally I don't trust anything but a prototype. I don't trust my own opinion, let alone anyone else about that something will be useful.*

On the other hand we heard the view: *We unfortunately used the word prototype the other day, but that's a bad word around here. It has a bad connotation - it implies that you get a half-way product and then it becomes the real product.* Clearly different cultures see prototypes in quite different lights.

Being a new technology, PCEs are more prone to being developed in an ad-hoc, exploratory manner.For this reason, a third interviewee emphasized that PCE prototypes have to be managed in a disciplined way if they are to be effective. Perhaps this provides an insight into the different views expressed above: *You need to interject some notions of functional decomposition and functional verification for each of the prototype levels, so that you can say "given the fact that I'm going to make this my prototype goal, I'll actually do some algorithmic development on paper, reason about it and then I can go deeper into the prototyping." In that way you can probably eliminate much of the code-and-go activities.* With respect to a disciplined approach, the same interviewee also made the following comment about managing prototypes: *We had the issue of having to manage prototypes as prototypes -- you'd better make sure of configuration management. That's a lesson we forgot a couple of times.*

4.3.5 Control Driven Versus Artifact-State Driven

Process end users need to feel that they are in control of their immediate work, not micro-managed by their supervisors, or unreasonably constrained by an arbitrarily-imposed process sequence.

This issue translates into whether the automated process should be driven purely by changes in artifact state, or whether the process should also allow for the explicit modeling of external control[1]. We found that interviewees generally had a desire to minimize the amount of overt, externally imposed control. One interviewee stated: *[Software developers] see changes between states. They don't think of this as process, but the natural progression of their software through the organization.* However, another interviewee suggested that overt control could not, realistically, be entirely eliminated: *Bob and I spent a lot of time talking to [an organization] and Bob convinced them that state-change architecture was a good thing. They got this idea that they could handle all process enactment by just modeling artifact states. There is no notion of process control – they thought that process control could all be derived from artifact state.*

The fear of an all-controlling machine was allayed in one case after the end user had a chance to actually get some hands-on experience. *One representative of a trade-union was on the team and he said that the tool was something like Big Brother, but he said this only once and*

1. E.g., a state-change driven approach implies that the new activity, *build*, can be initiated when a code module is transformed from the state *not-debugged* to the state *debugged*. The control-driven approach implies that a manager (or the machine) deems, somewhat arbitrarily from the end-user perspective, when it appropriate to start the build.

then forgot it, because the manager took care of the problem. The tool is very open to everybody and is not used to control people.

The concern over unnecessary machine constraints was voiced by another interviewee: *People, especially creative people, don't respond to a tool which says "you will start the activity now," for example you cannot start implementation until your low level design has been approved. They cannot work this way. It is better to define the artifacts that are to be produced and the goal states that these artifacts can be in. As a process definer, I can say: here are the exit criteria which I will impose on you under which you can officially declare that a goal state has been reached. As a project manager I have every right to impose these criteria on you. I overstep my bounds if I tell you to use this method, or you will start this process at that point, and not do so until you have finished something else.*

The same interviewee provided some further insights into this area. He indicated: *Typically a programmer has 20 things going on at once, none of which are finished. An engine that assumes things are serial or sequential will not work - it will last about 2 days.* In a similar vein, he stated: *End users may say [to the manager] "you didn't prioritize any of my activities but I wish you would. I have 30 activities and need help in prioritizing." But don't tell the end users what they have to do or it will be rejected.*

4.3.6 System Performance

Slow performance, indicative of immature systems, was a common theme. One interviewee indicated: *We had a problem that the environment was incredibly slow – we had an underpowered processor. We were doing a lot of processing so this added a burden to the workers who were trying to code as fast as they could on their projects but they found that they had to spend 15, 20 or even 30 minutes a day messing with process enactment stuff. We thought that was too much of a burden. One of the requirements of process enactment is that it can't make life a lot more difficult for the individual workers. If we want them to get through a few screens to get to their work then it can't take too long for them to do that.*

The same interviewee indicated: *Every time there was an actively running process, we would actually have two operating system processes running on the server. The server has a limit. These were all running off the same administration ID, not the worker's IDs. So there would be 50, 60 or 70 processes running, and eventually the system would hit a limit.* This problem has not been overcome in an improved version of the process-centered framework, but it does point out practical issues with implementation of large-scale processes.

Another interviewee voiced similar frustrations: *One of the things we are trying to do is to keep the tool very small on the client side because PCs just don't have the power, memory etc. FlowMark® took a big wad of disk space even for the client*. The same interviewee indicated: *Some of the difficulties we had were with tools. We had many people with 386 machines and didn't have large hard disks. Anything we had to do with X-windows emulator really slowed things down. There was a lot of network traffic. People don't want slow response times. You have to make sure that whatever solution you give them is going to fit into their environment.*

No way can people around here afford to go out and buy 250 Pentium processors, or X stations. That's not going to happen.

With respect to crash recovery, several interviewees voiced difficulties. In one case, thunderstorms frequently initiated system crashes. The interviewee stated: *Of course that was not only a burden of the administrator, but was a hassle for everyone. People would have to check files out, while in the middle of performing tasks and would have to figure out what the status of each of these tasks was.* With another system, crashes also occurred fairly regularly at first: *The system administrator didn't feel too confident in the architecture due to the implementation of the database, and the fact that in the beginning he had to reboot the system regularly. However, despite these problems, users were satisfied with the system response times and the reboot rate has decreased, currently to 5 reboots per month.* A third interviewee indicated similar frustrations*: There were problems with graceful recovery from power loss. Each time we needed to bring the system back up, an expert was needed to put things back in order.*

5 Conclusions on the Interviews

We originally set out to focus on end users but found ourselves talking more often to process-centered environment developers. The reason for this was that we found few software organizations using process automation on a day-to-day basis. We talked mostly to quite large government-funded efforts, and while we would like to have examined small "home grown" initiatives, we did not find many. To date experience with software process automation appears to be limited, but we did find some people who were struggling with many of the technical and non-technical issues pertaining to the adoption of process automation. These experiences were the subject of this section of the report. The following points summarize what we heard from interviewees:

- **On institutionalization:** We interacted with many pilot projects, but saw few successfully institutionalized environments. Many organizations were building prototypes and doing technology assessment. However, in only two organizations did we see automated processes that were institutionalized. These organizations were a bank (where process automation helped maintain their software), and at a commercial company (where their process-oriented CM product was used to upgrade this product).
- **On primary motivators:** Productivity and cost reduction were primary motivators, while quality and process improvement appeared to be secondary motivators. Because of high turnover (particularly in the military) there was interest in support for training and maintenance of legacy systems.
- **On maturity of the technology:** Process automation tools do not yet appear to be stable or mature. Some interviewees experienced frequent crashes with resulting restart difficulties. GUI builders were limited in some tools, while in others there was a limited ability to query information graphically. There were also some performance and network traffic problems.
- **On process definition:** Process definition was identified as being one of the most time consuming aspects of process automation. It requires a level of precision that is very time-consuming, and obtaining consensus can be a quite contentious process. The devil is in the details.
- **On external consultants:** We found that resistance could be high to process automation if it was perceived as being imposed by outside agents or consultants.
- **On application to software development:** Some people viewed themselves as innovative and viewed process automation as inhibiting this creativity. This was particularly true when process automation was applied in software development area, as software processes are often non-routine, and end users are highly educated. Software tasks tend to be of low frequency, and software processes vary too much from project to project.

- **On incremental strategy:** While few process automation projects had used an incremental adoption strategy, there was wide agreement on using this approach (as opposed to a "big-bang" approach). Many felt that an incremental approach should be used to produce quicker feedback, and that it was desirable to introduce a manual version of the process before making the transition to automation.
- **On control:** Interviewees raised concerns about the control issue. First, process automation might provide supervisors with a level of intrusive control that would be counter-productive (the issue being one of whether artifact states, or human intervention drive the process). Second, there was the issue of the automated process unreasonably constraining the sequence in which the end user can perform tasks.
- **On adoption practices:** Many of the adoption issues uncovered were not unique to process automation. For example
  - There is a need to foster buy-in at all levels.
  - Success criteria need to be stated up front.
  - Potential risks need to be identified early and reviewed periodically.
  - To encourage buy-in, there is a need for cross-organizational teams.
  - The differences in attitude of old-timers versus newcomers needs to be addressed.

Appendix A Definition of Terms Used in the Report

Automation tool *is the software product or language through which the process-centered environment is built and executed.* Thus tools such ProcessWeaver, SynerVision or InConcert are process automation tools. It may also be possible to build PCEs using, for example, UNIX scripts that are enhanced with process-oriented functions (e.g., to support communications, controls, and end user interface capabilities). Such augmented scripting languages may also be considered process automation tools.

Champion *is a person who is an enthusiastic promoter of a technology and is sufficiently senior to influence management decisions with regard to that technology.*

Development team *is the group of people responsible for implementing and transitioning the process-centered environment to the end users of the environment.* Such a group will certainly involve individuals with a technical background. It may also involve people who specialize in organizational change or specialize in the cultural aspects of technology adoption.

End user *is a person who produces product by interacting with the automated process.* Thus, for example, if the automated process supports document review, then a person who performs document review is a process end user.

First-line manager *is a manager who is responsible for a project or business group.* In general a first-line manager will not have other managers reporting to him or her.

Formal (or ***formally***) are sometimes used as qualifiers. Either word implies there is a recognized, documented, and agreed to standard to which the associated concept conforms.

Process automation *is computer-based support for the flow of work between individual tasks.* Processes (large or small) are said to be automated if manual control of task initiation or sequencing is transferred to the computer. It may be driven by a simple computer-based script or it may be based on a process-centered environment. Such assistance may involve only one person and one computer, or it may involve multiple persons supported by multiple computers terminals. However, for the purposes of this study, an essential component is human communication. Hence the automation shall allow communication between at least two people.

Process-centered environment (PCE) *is an computer-based environment in which multiple people interact under the management of a computer-based process, and in which there are well defined mechanisms for human and machine communications and control.* It is likely to be built using a *process execution tool* that has a process programming language with which to express process execution constructs. A PCE may, for example, enact a bug tracking process, a peer review process, a testing process, or a configuration management process.

Process-centered framework *is a software product that provides the functionality necessaary for the definition and enactment of process.* It includes a process enactment language, mechanisms for process enactment, a means to invoke tools, support for communication

(between persons and tools), and capabilities to support the construction and debugging of process models described within the process language.

Organization *is a set of projects or business groups that share that same basic corporate and technical cultures.* They may or may not work in the same product line, but probably share the same values, use similar technologies, and have open lines of communication. A project or business group is supervised by a first-line manager.

Senior manager *is a manager who is responsible for a number of projects or business groups and has financial responsibility for and control over these projects or groups.*

Appendix B The Interview Question Script

Introduction

1. Thank interviewee
2. Introduce team members
3. Purpose of interview
4. Confidentiality/attribution statement
5. Can we tape?
6. What the roles are
7. Review interview topics
8. Length of session, time constraints
9. Ask of there are any questions before we begin?

Business/product characteristics

Hypothesis: Business/product environment defines the types of processes and hence the effectiveness of automated processes.

General question: Describe the nature of your business

1. Who are your customers?
2. How large is your business unit and how is it organized?
3. How would you describe your organization's culture?

Process maturity

Hypothesis: One must be a process-mature organization to effectively use process automation.

1. General question: Describe any process improvement efforts in your organization.
2. Have you had a process assessment or capability evaluation? Please describe.
3. Do you have a process improvement plan in place?
4. Do things usually run smoothly when developing your products?

5. How do you plan and manage projects?
6. Do you use a CM system to manage your products?
7. Describe project management or product metrics that you track (if any).

Application focus

Hypothesis: some types of process are more appropriate for automation than others.

1. General question: Describe the process are you automating.
2. Why did you choose this particular process to automate?
3. What benefit do you expect to get from automating the process?
4. Were the processes that were automated typically ad hoc prior to automation?
5. Describe any metric data you are collecting automatically.

Use of tools and technical development

Hypothesis: A familiarity with technology, prior to the automation project, is a prerequisite to successful automation.

1. General question: Describe the technical development of the automated environment.
2. General question: Describe what tools and technology you use.
3. What tools did you use to construct the automated environment?
4. How did you select the tools (to support both process and applications)?
5. How were integration issues (tool/data/control/process) handled within the environment?
6. How effective were the mechanisms for constructing the automated processes?
7. Does the environment often crash?
8. In which applications, if any, does your organization use CASE tools?
9. Have you used any CASE tools – independent of the automation activities?
10. Has your organization performed any CASE-tool integrations?
11. Are there other technologies that you have used on a trial basis?

Team characteristics and experiences

*Hypothesis: A range of both technical and non-technical competencies is required to implement **process automation technology.***

General question: Describe the process automation team and their backgrounds.

General question: Also tell us about the end users and their backgrounds.

1. Were the end users involved in defining the automated processes?
2. Tell us about end user experiences with using the automated process.
3. Was the automated process perceived as being too constraining on the end users?
4. Did the end users get training in the automated process?
5. Did the automation team get training in the automation technology?

Transition and adoption

Hypothesis: A well thought-out transition/adoption strategy is critical to end user acceptance.

General question: How are you introducing process automation?

1. Who sponsored the automation and how serious was the sponsorship (e.g., funding)?
2. Is anyone perceived as the main driver (champion) of the process automation project?
3. Describe your implementation plan.
4. Did the adoption use a all-in-one approach, or an incremental approach?
5. Do you have someone responsible for maintenance of the automation process?

Impacts and insights

Hypothesis: Successful use of process automation should correlate with the capabilities identified in the six areas defined above.

General question: What impact has process automation had in your organization?

General question: Describe your future plans for process automation.

1. If you could start from scratch again, what would you do differently?
2. What were the most technically challenging issues in developing the automated process?
3. What were the most socially challenging issues in developing the automated process?
4. Describe tangible benefits of implementing process automation in your organization?
5. With which applications do you think process automation can be most effective?

Wrap-up

1. Are there any other things we haven't asked you that you think we should know about?
2. Do you have any questions about the study?
3. OK if we call you for clarification or additional information?
4. Review any action items (e.g., requests for info)
5. Reassure confidentiality/non-attribution
6. Thanks

Appendix C Summaries of the Interviews

C.1 Project A

Business/product characteristics

The organization for which the process automation system is intended is responsible for the maintenance of a large number of military software systems. These systems are widely diverse in type, (from management information systems to command and control), size (from 50K to 1.5M source lines), and language (Ada, C, specialized military languages, and various assembly languages).

The organization is structured as a matrix, with personnel assigned from various specialty areas (e.g., testing, CM, implementation) to work on a particular application. In addition, a core of personnel are not matrixed, but rather are permanently assigned to the larger applications in order to provide focused expertise and continuity.

The personnel who are the intended end users of the process automation capability include a mix of civilian employees, enlisted personnel, officers, and contract personnel. However, the process automation efforts are dominated by military personnel.

The organization is currently subject to downsizing due to military cutbacks. However, the volume and complexity of software maintained by the organization are not expected to decrease. Staff reductions have been accomplished primarily by natural attrition. As personnel are transferred out, they are not replaced with new personnel.

Process maturity

The organization has a strong commitment to continued process improvement, following the SEI Capability Maturity Model (CMM). An initial internal process assessment was performed circa 1992. At that time, the organization measured at CMM Level 1. A large number of changes were instituted in a concerted effort to improve the organization's process capability. Among the changes were the creation of a Software Engineering Process Group (SEPG); the development of a group-wide software quality assurance and standards team; training in processes and methods; and the development of guidelines for milestones, schedules, deliverables, software specifications and requirements, and metrics. In addition, the effort to develop a process-centered environment had its roots in this initial assessment.

The SEPG is responsible for the overall process improvement effort, but has not historically been directly responsible for the process automation effort. The relationship between the process automation work and the SEPG developed late. Collaboration between the two efforts was limited until the late 1993-1994 time frame. The process supported by the automation effort was developed over a three-year period primarily without input from the SEPG. This process was developed primarily from a "grass roots" understanding of the processes in place

at the time, and it proved particularly hard to develop consensus due to the wide range of processes used within the organization.

There has been a fortuitous match between the tool and process support selected by the process automation group and CMM Key Process Areas (KPAs). The interviewees suggested that as the effort was initiated to climb the CMM ladder, the process automation work (particularly that involving configuration management) became important, since it addressed Level 2 KPAs. At the time of the interview, there was an ongoing effort to synchronize the process automation and SEPG work further.

The organization continues to actively improve its software capabilities, and in 1994 achieved CMM Level 2. The members of the organization with whom we met expressed pride in this achievement, and suggested that they had come a long way in the ability to maintain software.

The organization is currently collecting metric data on Source Lines of Code (SLOC), Software Problem Reports (SPR), Change Requests (PCR), and effort (time tracking). While the data is collected, it is currently not analyzed and used to any large degree. However, it was reported that metrics collection is becoming increasingly important in order to justify manpower. Without "hard" numbers, managers expect to have increasing difficulty in justifying staff.

Application focus

The impetus for the process automation activity was a particularly large and complex Command and Control system that was to be delivered for maintenance. The original effort involved developing a support process for this system, since the anticipated maintenance staffing was expected to be inadequate without significant improvements to the process and tools.

In order to justify the effort to build a process centered environment, advocates developed an extensive business case analysis (circa 1993). An estimated return on investment (ROI) was calculated based on historical data from maintenance of similar software, Barry Boehm's model of software cost drivers, and input from a consulting firm. However, the estimates appear to be based primarily on returns from use of CASE tools for static analysis and reverse engineering activities, and not on an additional return due to process automation. Regardless, the business case analysis provided optimistic estimates for an ROI of 17 times initial investment at ten years, with a break-even point occurring during the first year. The business case analysis also suggested that staff could be cut by 50 percent through the use of process automation.

Over time, the mandate of the process automation group has expanded to include process automation support for the wide range of systems maintained at the facility. This expansion of the mandate has proven to be particularly troublesome. It has been extremely difficult to reach consensus on a common process, due in large part to the varying maintenance processes of the different groups. The initial "common" process definition involved 13 highly detailed steps. The current common process incarnation involves seven less detailed (higher

level) steps. However, achieving consensus on even this more general process has proven difficult.

Use of tools and technology

The tools included in the environment are

- CCC Manager for CM
- A custom (military-language specific) parser for the COTS Refine/C program analysis/understanding tool.
- Refine/C. The Refine/C tool provides coding standards checks, complexity analysis, call graphs, and cross-reference information
- SoftBench C, providing static and dynamic analysis
- DDTS, a requirements tracking tool, that is being used to handle change requests and problem reports.
- Autoplan, a project management tool
- Framemaker, a document processing system
- Worldview, a document viewing tool that can create hypertext links
- Synervision

SynerVision from Hewlett Packard is used to control the sequencing of user activities and the activation of tools. However, not all tools are connected to it; Refine and Autoplan remain outside of the automation effort.

The emphasis within the process environment is on control and presentation integration via SynerVision (and HP's Broadcast Message Server, which is the underlying technology). Synervision provides a user interface that identifies tasks, provides access to tools, and displays process information.

Data integration has proven to be a difficult problem, particularly when the data from various tools "overlap" (i.e., data that are essentially the same are stored in multiple tools in different formats). The process automation developers reported that some of this problem can be dealt with by very careful encoding of the process and creation of appropriate "triggers" to automatically update data in various tools, but this approach is difficult and at best a partial solution. The process automation developers considered developing a richer set of relationships between data by incorporating the PCTE object management system, but rejected this solution because of lack of vendor support for PCTE.

The process automation developers report that actual "coding" of an automated environment with SynerVision is not difficult, although there is a perceived need for a better process

specification language (SynerVision provides only very low-level language for encoding process). However, the "engineering" (incorporating specification, design, integration, etc.) of a process environment is more difficult. Three problems were identified: 1) definition of the process is particularly difficult 2) tools do not have adequate application programming interfaces for the level of control and data access required for deep process automation 3) Seemingly minor changes to the process can have a major impact at the detail level of the process encoding.

The developers reported a number of environment reliability and performance problems. In characterizing reliability problems, they termed the environment as somewhat "brittle." For example, adding a new user to the system caused a flaw in the process environment that delayed an important demonstration. They also reported problems keeping abreast of changing tool versions. The performance problems reported were attributed in large part to the characteristics of the network on which the system ran rather than the process automation itself. This indicated to the developers that the use of the process environment might place additional constraints on the underlying computing environment.

Team characteristics and experiences

A group of eight individuals working for a large military maintenance organization were interviewed. The interviewees included a mix of civilian and military personnel involved in the development of the process automation system. Two individuals were members of an SEPG. No end users of the automation were interviewed, although it is reported that there were some end users of early versions and components of the system.

The process automation effort has consumed approximately 4 calendar years and 18 person/years of effort. Staffing during this effort has involved two to six integrators/toolsmiths. However, the majority of the effort expended has been in process definition. Most of this work has been thrown away.

The process automation team is primarily comprised of military officers. This has proven to be somewhat of a problem, since the actual team composition has changed due to reassignment of personnel. One critical change in the team has been the loss of the original, strongly supportive sponsor. Team members felt this had a negative impact on the team, since more time and emphasis had to be placed on preserving organizational support.

The team has two main parts: a group defining process, and a second group developing the automated environment. Members of the team identified three essential skills for team members: 1) process definition, 2) process automation tool expertise, and 3) specific expertise with each of the integrated tools. It was generally lamented that the team lacked expertise in a fourth skill area: technology transition and adoption.

Significantly, as previously indicated, the process automation team did not collaborate with the SEPG until quite recently. It is not clear what effect this had on the team's ability to define a consistent and acceptable process or to prioritize work.

Transition and adoption

The process automation team admits that transition of the technology has been a particularly difficult task. They suggest that their initial lack of knowledge about transition problems and approaches has worked to their disadvantage. While individual tool training has been conducted, the automation group spends much of their time providing "hand-holding" in order to facilitate tool use. This "hand-holding" approach has led to moderate to good organizational acceptance of a number of the individual tools, including the CM suite, WorldView, and Refine. However, they have not developed strong transition/training plans for the process automation component.

The interviewees reported that while support for current tool users is not clearly defined, process automation personnel are using any opening involving support for end users to get them to accept larger pieces of the automation. Currently, end users are supported through "gentlemen's agreements" with process automation personnel. The general feeling on the part of interviewees was that this very personal, direct support approach is successful, but is very demanding of time and resources. The interviewees also reported that this approach came about because one report suggested that the process automation group was not sufficiently focused on the actual user problems.

The process automation personnel have also come to recognize long term maintenance of the environment as a significant problem. The group spends significant effort just to keep the environment working. They suggest that a core group will be needed to plan upgrades, enhancements, and maintain the environment. The process automation group is hoping that the SEPG will pick up support for the environment.

An interesting footnote to the adoption planning involves the support being provided by an external consulting organization. This consulting support and cooperative effort was expected to lead to a process-centered environment that would be marketed by the consulting organization. However, no marketable environment has emerged, and the interviewees felt that it was possible that plans had been changed.

Impacts and insights

The process automation personnel report that the most vexing problem they have faced involves the identifying of a process that was acceptable to the wide range of expected end users. Process identification has consumed the majority of time and resources on the project; the actual automation of the process and integration of tools has been far less difficult. In order to achieve consensus, a greatly simplified and less involved process is to be supported by the environment; even this process is open to debate. As a result of their problems with process definition, the interviewees suggested that organizations wishing to develop an automated environment should focus on identifying a firm set of requirements for an existing (and accepted) process; attempting to define a new process with potentially conflicting requirements from a range of end users is quite risky.

In addition to the hard-learned lessons concerning identification of an appropriate process for automation, the interviewees provided a number of other insights. These include:

1) The organizations with the best-defined processes prior to automation were most receptive to automation and tool support. The interviewees reported that the testing group, which had a particularly well-defined process prior to the development of the process-centered environment, reported saving many hours of labor due to the use of the tools. In addition, this group was an early adopter of the process automation component. On the other hand, some groups with poorly defined processes viewed the automation effort as a way to "force" process on them.

2) While the overall technical challenges involved in process automation were relatively few, data integration was a major exception. Careful encoding of the process can provide some help in ensuring data consistency, but many tools do not provide sufficient access to data to support this approach.

3) Providing the tools to be used in the automated environment prior to providing the process automation component can be a useful strategy. This gives the end users some experience and acceptance of the technology, and helps them define the process integration requirements. The interviewees suggested that as end users became more sophisticated in their use of the tools, their integration demands changed, and often increased.

4) High-level sponsorship must be identified and nurtured, perhaps through the release of intermediate products that can regenerate enthusiasm. Interviewees suggested that the lack of a substantial intermediate product has led to the loss of support for the effort.

5) Process automation advocates should beware of flashy demos that "sell" the product, but do not reflect the reality of the process integration. Such demos lead to unrealistic expectations, and subsequently to disappointment.

C.2 Project B

In this interview there were two interviewees, a manager and an engineer. While the manager provided most of the information, the engineer support this information with additional (and sometimes revealing) insights.

Business/product characteristics

The organization develops aerospace software applications. It is supported by a staff of about 260 people.

Process maturity

The organization is rated as one of high process maturity. This high maturity level was motivated by the need to produce software whose defect rate was extremely low. However, this high software quality was produced without the support of sophisticated technology; their tools

were relatively primitive and the high level of maturity was a result of considerable manual labor. Many of the engineers had been working for years with old technology (e.g., dumb, mouseless terminals), and found the transition to PCs intimidating and resisted these changes. However, because the organization was comfortable with working within a defined process, transitioning to automation was not so difficult in that regard.

Because of cut-backs in federal funding, there was a need to produce the software more cost-effectively and process automation was seen as a technology that could support this need. However, the decision to use process automation was not an explicit requirement of the funding agency. The latter only stipulated that cost reductions had to be made.

Application Focus

The aerospace software applications that the organization deals with include guidance, navigation, and flight control. The organization is responsible for developing, analyzing, testing, and simulating the use of this software. Simulators support the analysis and testing. Many languages such as assembly, C, PL1, and Fortran are used.

Use of tools and technical development

CM is performed through one of two database products - IMS and Oracle rather than an explicit CM tool like CMVC. Requirements traceability is all done on paper. Individuals each own pieces of the process, and they are responsible for enforcing it. There is a lot of manual checking.

Currently the organization is not using any CASE tools. They were experimenting with Software through Pictures and InterLeaf. However, there was never any serious use of these products. Some of that has to do with the lack of support for specialized language that were used (such as HAL), and some of it had to do with incompatibilities of the hardware platform and operating system that were used to support the CASE tools relative to the commonly used platform and operating system.

The organization supports the beginnings of a process asset library. The inspection process is one that is currently reused frequently, because code inspections, design inspections, and requirements inspections only differ in the roles and the inputs. The organization is struggling to figure out what other elements are reusable. The interviewee was concerned, for example, that testing processes were very different from development processes. However, the interviewee also felt that many of the projects perform very similar activities; they just use different acronyms. He thought that the key was to get buy-in from an initial team of people who would spread the word.

The interviewee indicated that they had used both ProcessWeaver and FlowMark. However, they were currently building their own process automation development tool. While both ProcessWeaver and FlowMark had their limitations (e.g., he felt that neither was particularly easy to learn) the major motivation behind building their own tool was that money was available for this, while it was not available for the licenses to purchase the commercial tools in sufficient

quantity. The interviewee indicated his frustration with this by stating: *Why reinvent the wheel....Anything we build can't match what people have spent years building and developing.*

Team characteristics and experiences

In earlier years, the organization was well funded, had a stable engineering staff, and was supported by technology that, while relatively unsophisticated, was understood by all. More recently, funding has become much tighter, and the organization is now operated by a different parent company. Changes such as these have resulted in more frequent personnel turnover, with tensions produced between the older staff and the new-comers. For example new-comers are much more open to the introduction of new technology. As the interviewee stated: *Not all old-timers are against this stuff, but there is resistance by people who have been here longer.*

Transition and adoption

The interviewee pointed out that long-term employees are often reluctant to change to new technology. While this is true in general, it is a particular problem with process automation as this can be perceived as being particularly intrusive. On the other hand, less experienced engineers are more open to the support and guidance that process automation can offer.

Another identified issue that inhibited the introduction of process automation was job-security. This was particularly so among coordinators, whose jobs are most closely associated with managing process. In addition, partially from job security-related concerns, there is resistance to share information. For example "old hands" were reluctant to share knowledge of cryptic commands in the belief that such sharing made "old hands" less needed. However, with older staff leaving and new people coming on board, the interviewee felt that it was essential to capture the knowledge before it disappeared. As he stated: *Management is pushing the sharing of information because you do not know how long you are going to the here, because you may leave tomorrow. So this is slowly going away. Not having time was a good excuse to not share. If there was a request to share, then you might be shown the manuals (the what) but not given any verbal support (the how).*

Resistance to automation can even come from attempts to introduce new screen displays. The interviewee explained that in automating one process, the automation team created the capability to move the data from Oracle into Lotus Notes. However, the people in the verification project never wanted to see a Notes screen – they hated it, because it was not what they were used to in their group. So screens were developed using Dr. Dialog (a GUI builder). When the developer people saw a demo of this, they said "I though you were going to call up Notes for us – we don't like this" As the interviewee expressed: *Those are the problems that you have. It's "I made this, and its mine and if you say bad things about it, I don't like you." And when you try to explain that there is a better thing than this or something that can make it better, it's really hard.*

One problem encountered was that the managers were not technology-oriented – most did not have PCs at home – so that explaining the nature of process automation was a challenge. However, the interviewee expressed the importance of keeping management in the loop, and making sure that management were accounted for in the defined processes. The interviewee felt, nevertheless, that managers should not be part of the process definition activities, as managers tend to be inhibitors.

The interviewee felt that the need to involve all organizational groups in decision making was a critical success factor – in particular, those who may be hostile to the technology need to be included. In the experience of the interviewee, one antagonist became so involved in the process definition activity that he became an enthusiastic supporter, spending weekends in process-capture activities. This lesson was learned the hard way. Previously, people had been ignored. As the interviewee stated: *Now we say "what bothers you." With the representatives of each different area in agreement, we can take that agreement back to our management and to our own people and convince them. There has to be a lot of selling. Back in the past, you convinced your own area to support your pet product, and then you find that all the other areas are not going to embrace you because you did not include them.*

The interviewee found that it can be difficult, even for technically-oriented staff, to understand the concepts behind process automation. This problem seemed to be true both with new and experienced people. Even after demonstrations of the capabilities of ProcessWeaver or FlowMark, confusion was evident. One reason for this confusion appeared to be that the automation tool did not produce any product as opposed to a compiler or an editor. Another reason was that any demonstration was likely to show a process different from those with which the audience is familiar. They did not realize that the tool can be used to model arbitrary processes. A conceptual problem with confusing terminology also arose; for example, in the difference between process automation, workflow, and groupware. The interviewee felt that these issues point strongly to the need for basic education in the fundamental concepts as well as applied training in the technology.

One example highlights the problems of communication and technology mismatch between a technology producer and the consumer. At one point, part of the organization's management were interested in transitioning a process-centered environment from another division of the company. This environment had a relatively fixed, built-in process. In the interviewee's words: *[The technology producer] provided us with Cadre, TeamWork, FrameMaker, something for requirements traceability, their own homegrown scheduling tool, and their own homegrown tool integrator. They convinced our management on this new project to use the environment for Ada code development. What happened was that they had DEC machines and not Suns, so they had to port to the Sun. People started using the automated environment but they didn't like the process that was enacted in it. It was not their way of building software, since it was brought in from California. There were lots of meetings and phone calls – we need this, we need that... The whole function of the [technology producer] for a year was – you'll do nothing but make [the technology consumer] happy. This was perceived here as, too little too late. What our people said was – we really like Cadre. They pulled these tools out*

of their unified environment, and they threw away the rest, but didn't tell the folks in California, they just kept giving them more requirements. So California kept sending tapes and bringing people to load the system. They'd go through it, check it out and leave. Then they'd put the tape on the shelf. It was interesting that people basically said "That's not our process and it's way too slow" They could not handle that. They took out the "best of breed", and didn't communicate back to the folks back in California that was what they were really doing. When those people found out about it they got all upset – we've done all this work for you guys and you just put it on the shelf. And we said – you never asked us the right questions. It was like two trains coming down the track in opposite directions.

One final adoption issue that came across clearly was the need to put process technology in perspective, placing emphasis on the process first and introducing the technology support incrementally. Again in the words of the interviewee: *[Y]ou must learn your process before you enact. Process comes before tools. We are very strong over that. A tool is a tool... You can't throw a switch an enact a process. Tools should be chosen to match your needs... Capture your as-is process, re-engineer it slightly to take advantage of any tools on the computer, have a review, model your process, have another review to make sure you did that right, enact it, simulate it, review it, then integrate your tools and forms and review it, use it then repeat.*

Impacts and Insights

The interviewee was asked if he had to start all over again, what would he make sure he did. In response he identified a series of issues that, because they were so articulately expressed, are listed almost verbatim below:

- You have to have an across-the-organization functional team to get everyone excited about doing this. Ultimately if a single organization comes up with something, it's looked on as an anathema by the other projects.
- You need to define very crisply what your goal is, and not make it an ambiguous thing such as: you are going to automate the corporate personnel process.
- Start by having little pilots - that's very important. On the project, we started with a small subset of the people developing requirements. In less than a week they said, "the people over [in project X] now want it for their department and the people [in project Y] want it for theirs" and so it started to catch on.
- You really need some technology advocates – people that can go proselytize it out to the organization. People who are trusted in the organization. It's easy to get someone who wants to get on to your band wagon but who does not interface well with the group. So when you form your team, get someone who gets excited about it, and can go back and spread the word.

- You have got to come in from the top. You have to convince upper management, because they are the ones who say the dollars are OK. You've got to get these people – they are by far the hardest – certainly in our organization. Our contractor gives us lots of money, and we do lots of good things with that money.

- Some of the difficulties we had were with tools. We had many people with 386 machines and who don't have large hard disks. Anything we had to do with an X-window emulator really slowed it down. There was a lot of network traffic. People don't want slow response times. You have to make sure that whatever solution you give them is going to fit into their environment. There is no way people around here can afford to go out and buy 250 Pentium processors, or X-stations. That's not going to happen.

- The Unix-word scared the heck out of everyone because we had been IBM users, and IBM didn't speak the U-word for many years. And we were there with IBM Unix machines (AIX). So don't come into an organization and completely undermine its heritage.

C.3 Project C

Business/product characteristics

This interview was conducted in the same organization as Project B. See Project B for Business background.

Process maturity

This interview was conducted in the same organization as Project B. See Project B for process maturity information.

Application Focus

The interviewee was involved in developing two process-automation related systems, one to track project requirements in a system engineering environment, and the other to support distributed project management. Both of these systems were built based on the process automation tool ProcessWeaver. In the latter case, ProcessWeaver provided support for coordinating the process of reconciling loosely coupled schedules across departments. The basic functional components of the system are: project management (including time collection), database management (for metrics support), and publication (for report generation etc.).

Use of tools and technical development

The major integrating tool used in this project was ProcessWeaver (from Cap Gemini Sogeti). To support the functional aspects of the process, Schedule Publisher (from Advanced Man-

agement Solutions), Oracle (from Oracle Inc.), Interleaf, and WorldView (both from Interleaf Inc.), Open Interface, and Data Access Element (both from Neuron Data) were used.

Team characteristics and experiences

This interview was conducted in the same organization as Project B. See Project B for team characteristics.

Transition and adoption

Contrary to what we heard from the first interviewee, the second interviewee did not notice any clear difference in the acceptance of process automation with respect to duration of employment. He did feel, however, that there was a difference in acceptance depending on roles and responsibilities. He believed, for example, that there was more reluctance on the part of first-line supervisors because their immediate schedules prevented them from looking too far ahead. For first-line managers to be supportive, he felt that they had to see what was in it for them. Second-line managers suffered less from this problem.

The interviewee also indicated that it could be difficult to convince management to invest money in process automation if they did not see a direct value in it. He felt that one may need a champion to support the technology if it is to succeed. In this regard he felt that engineers with systems or project management backgrounds tended to make effective champions. Talking about a champion in his area, the interviewee indicated: *He was one of the senior engineering types in the systems engineering organization, but he was not a manager. In fact, his manager may have been one of the stumbling blocks but that managers' manager was a proponent. I knew [the second-line manager] pretty well and had worked with him for a long time. So we were able to work around the first-line manager.*

Impacts and Insights

The interviewee provided some insights into how people with naive notions about process definition proceed to describe their processes. In the interviewee's words: *What they had was a process, but when we asked them to write it down, they did so in what they thought was a very detailed fashion. When you started looking at it you would come to dead ends in the process. When you asked about that, they'd say – well, we sometimes we do this and sometimes we do that. We told them when you put in into a computer, you have to state this way or that, or flip a coin. They had a process, but in some cases it was not thoroughly defined– to the point that you could enact it in a computer. Just the idea of having to code the process into a computer caused them to sit down and define the process to the level that the computer needed.*

An issue that arose was – what processes are appropriate candidates for automation? The interviewee offered two criteria that were important to him. The first was to select a process associated with a group of people who were willing volunteers. The second criterion was to select a process that was meaningful, i.e., a process that really supported getting the job done.

One issue that came up with regard to defining processes was – how do you know when you have or have not arrived at the correct level of process granularity? This question was asked of several interviewees, and none came up with clear criteria. However, this interviewee did feel that it was important to involve someone relatively senior to provide guidance. As he stated: *We contended that enacting process was like critical path scheduling. You can generate a schedule to such detail that you reach a point of diminishing returns. Likewise in enacting a process, you can go to a level of detail where it would be a burden more than a help. You need a sufficiently senior person to capture the process in order to decide how deep or detailed you want to go. The tool can do anything you ask it to do, but you do not want to excuse yourself to go to lunch.*

C.4 Project D

There were three project personnel interviewed during the interview session: one captain and two civilian employees.

Business/product characteristics

The organization maintains software systems for an agency of the Department of Defense. They support the software that performs command and control functions such as surveillance and ground tracking. The demonstration project is specifically devoted to a mobile application for command and control. A second command and control system that has very high priority in the short-term is also under consideration for processes automation. One of the interviewees felt that was an endorsement of the power that they had already demonstrated. Their main organization said: "OK, R&D branch, you've been talking about this stuff long enough, now prove your capability."

Process maturity

One interviewee felt that having defined processes was critical to organizations like theirs because there was such a high turnover of personnel. With no defined processes, he felt that there was a constant need to rebuild. He had been on the program for two and a half years and almost everyone was new. There was a going away party every week or two.

At the same time that Project D was trying to figure out process automation, they were trying to figure out how to implement effective processes in the first place. The rules the project learned through this experience were: 1) you have to have a widespread understanding and agreement on the foundation of your approach 2) and that it can be compartmentalized into different categories. One interviewee felt that once the approach is understood, you can then talk about the process details, making them methodical and systematic. He felt that, as implied by the SEI's Capability Maturity Model, some experience in trying a process is desirable, in order to make it repeatable before you proceed to refine it. He stated: *I've really come to*

understand how important it is to get to level 2. After you have defined your approach and process, then you can address process automation.

The interviewees' project was part of a larger organization that had the initial assessment about a year ago. That organization was assessed at level 2

Application focus

The target application of the process automation was a mobile command and control system that used a truck as a platform. The project was divided into several large activities and one of these activities focussed on process support. The system was coded in Ada.

Use of tools and technology

During the time that the pilot was being implemented, there were frequent lightning storms that led to system crashes on a daily basis. That turned out to be very troublesome, particularly in the complex multi-user environments that keep track of everyone's work. The system was unacceptably slow, as the processor was underpowered. Since a lot of processing had to be done, this added a burden to the software developers. They found that they had to spend as much as 30 minutes a day addressing problems with process enactment. They felt that this was too much of a burden.

Performance problems occurred mainly between the task execution tool and another tool that supported task dispatching. These were the components that were being most frequently used by the 12 people who were involved. Fortunately there were many patient people on the project, including the manager of the coding team. The interviewee indicated that most other managers would have shown much less patience. The end users were also willing to persevere while the development team solved the problems.

The task execution and task dispatching tools were originally designed so that they would talk to each other – when a small amount of information had to be passed back and forth. However the big data integration problem turned out to be between a process planning tool and the dispatching tool. These had totally different views of process (hierarchical vs. sequential).

Currently, in the task dispatching tool, one can get detailed descriptions of tasks. However, the development team is currently extending this so that one can view the current task highlighted in the center of the page with the other tasks that feed into and out of that task. There is also a *to-do* list. When tasks are checked as complete, other tasks appear. This is being extended so that one can look at future tasks, based on what we know now. It will do a best guess, based on predicted durations of future tasks. If task completion is late then subsequent tasks will get delayed. Currently the environment does not invoke tools. One of the interviewees felt that, while tool invocation was a small step, adding artifact management with its implications for version control was a really difficult problem.

In summary, the process automation project tried to do too much. There were some problems with configuration management, performance, crash/recovery, and the user interface. There

was limited funding, although the project was fortunate to have a lot of good will from the end users.

Team characteristics and experiences

On average there were about nine technical people on the development team although this number has risen and fallen. There were also a few managers involved. The project received a lot of managerial scrutiny.

One interviewee indicated that end users were open to automation and they were technically oriented. While there were some problems in areas such as configuration management and tool invocation, end users liked these features, even although they were not implemented very effectively. Prior to automation, there was an attempt to implement the automated process in a manual fashion. However, while the end users were very cooperative in defining the new process, they went back to their old process ways rather than enacting the new (manually implemented) process. So the manual implementation of the process was not successful.

Transition and adoption

An interviewee stated that one of the requirements of process enactment is that it cannot make life more difficult for the individual workers. For example, if the end users have to bring up several screens to get to their work then this operation has to be fast.

There was an apparent difference in philosophy between the civilian interviewee and the captain. According to the civilian interviewee, the captain would have preferred a screen that said*: Here is your task. You need these three tools. We have provided a button for each one of them. If you click on a button you will get the associated tool. In conjunction with the tool, you will need these four files to do your work, and if you click on those four file icons, the files will automatically appear.* In other words, the captain would like to bring the defined process down to a low level of detail. In the view of the interviewee, this could require elaborate configuration and version management, and someone would have to figure out, for example, that a certain abstract artifact translates into five files. Because of this, automated support will initially provide only a description of what the end users have to do.

In discussing the appropriate level of process detail, interviewees did not appear to have any strong feelings. One suggested criterion was the length of time a task takes. For example, the shortest task might be an hour. Another guideline suggested by one interviewee was that tasks involving two or more persons, who are doing separable activities, should be broken up. The interviewee did not want someone to be given a task that, when completed, is checked off, disappears from the task list, and a new task appears. In his opinion that was too detailed.

Impacts and insights

One interviewee indicated that some legal issues may be associated with the use of time cards. He provided the following scenario.

The automated process records how much time is being devoted to each task, and enters that data into a cost model, where the tasks are actually assigned to work packages. Meanwhile the same people are entering billable hours into time cards that the government uses to pay personnel. That results in two sets of books.

The interviewee noted that it is possible for a project to use only the process system to track time. However, he noted that, with every professional, the day is full of overhead activities that are not simply the tasks that are assigned.

There are payoffs to being process driven. The interviewee provided this example:

We used to have an astrophysicist who was very freewheeling in his approach. He was very resistant to doing things in a process-oriented way. This prima donna had to work with other people and they had nothing like the background in astrophysics that he had. Consequently, he discovered that, with process, he could communicate with these people, and they were able to code up his complex ideas with a lot less debugging.

Finally, an interviewee felt that process automation support is critical once the process is defined beyond a certain amount of detail. He felt that with a manually implemented process, it is very difficult to manage the details that can be managed using an automated system.

Miscellaneous insights from the interviewees

- In order to minimize the number of integration problems, it would have been better to minimize the number of subcontractors (under the main subcontractor).
- A more incremental approach to adoption of process automation should have been used. There is a lot of detail that had to be defined in order to make the process enactable. That consumes a prohibitive amount of time.
- IDEF0 is easily misunderstood by people enacting processes.
- Choose process support technologies that are proven and stable.
- There is the need to manage the changes in introducing process automation.
- Sometimes automated capabilities were being developed while concurrently supporting end users. That led to problems.

C.5 Project E

C.5.1 Interview 1

We interviewed the Vice President of Process Technology for a vendor of a major configuration management product. This product has significant capability to customize the

CM process. The interviewee's primary responsibility is managing the adoption of the product with the company's customers.

Business/product characteristics

This main mission of the organization is to develop and sell a COTS configuration management system. This software has a built-in process modeling capability. However, the company also provides products and services that help their customers define, refine and improve their processes. This interview focused on the issues around the adoption of their CM software and supporting process.

The interviewee's first goal on taking on this position was to develop an effective suite of adoption services for her customers. To this end, she spent 13 months visiting customers to identify their needs, and understand what adoption services should be offered. She then packaged that knowledge into a set of services. This package included items such as effective technology transition, process implementation, and requirements analysis.

The interviewee stated that the majority of their customers came to them after trying to implement a similar piece of software and failing. She indicated: *On the product side people reach a pain level where they realize the quality of what is being given to their customers is poor. They are in so much pain that they need to make a change. People see getting a new tool or product as a silver bullet. Then they realize that it's not a silver bullet.*

Process maturity

The process maturity of the CM vendor company was not discussed in any detail; the focus was on issues of clients process effectiveness rather than that of the vendor's organization.

The interviewee felt that they won contracts because of their process support, both for their tool and services. The interviewee stated: *The product comes out of the box with a process model -- which is generally reflective of the way that people want to do CM. It's the best way to do CM.* This product vendor has taken a generic configuration management life cycle process model and incorporated it into the product. She felt that this provides customers with an excellent base for becoming immediately effective in the use of the product and creating a starting point for adoption.

Application Focus

The application focus is configuration management. This CM product provides a basic process model that can be tailored by customers. The interviewee facilitates and leads the customer through a phased implementation approach aimed at reducing adoption risk. This includes planning, setting goals, and identifying changes to the configuration management process.

Use of tools and technical development

The CM system developed and marketed by this vendor provides extensive capabilities to manage software products. Thus it allows for the management of software objects, provides workspaces for the insulated development of the software's components, provides capabilities for software builds, and for the tracking, for example, of change requests. From our perspective, its most important feature is its ability to model sequences of activities in order to enforce the state changes that a software object may undergo.

Team characteristics and experiences

As explained by the interviewee, a typical customer adoption team combines both customer and vendor resources – vendor resources help assure buy-in while the vendor experience helps with implementation. A set of templates supports the adoption model summarized above. The customers do not generally appreciate the templates until they move into the adoption phase. The templates also provide the customer with the knowledge needed to deal with troubleshooting.

Transition and adoption

To help facilitate the adoption effort, the company developed a "typical adoption model." It consists of several phases as summarized below:

- Planning and Analysis Phase
  - Get management buy-in.
  - Gather and prioritize requirements for all user groups.
  - Write adoption and risk management plans.
  - Identify adoption team.
- Organizational Process Definition Phase.
  - Define and document as-is and to-be CM processes.
  - Implement process model in CM tool.
  - Pursue adoption activities (addressing risks, training etc.) with adoption team.
- Pilot Projects
  - Choose pilot team members.
  - Define project success criteria.
  - Document implementation plan.
  - Do pilot projects.
  - Evaluate pilots.
- Roll-out
  - Define incremental approach to organizational implementation.
  - Get buy-in for organizational roll-out.
  - Prepare end users and address resistance.

- Do implementation

- Process Improvement
  - Do post-adoption review.
  - Recommend process improvements.

The interviewee believed that many of the problems with successfully implementing a CM product had little to do with the actual product. She felt that many companies did not know how to deal effectively with technology change, and that companies do not adequately understand the adoption process and are not willing to commit to managing adoption as a project itself. Thus the resources, time or money, are not provided to make the adoption succeed. As a result, companies generally fail in adopting new software tools before they realize they need assistance with adoption.

The interviewee suggested that a key to successfully adopting the product is setting the correct expectations throughout the adoption cycle. Therefore, if the first three of the above phases are understood and followed, the change process becomes much easier by the time the rollout phase is reached. Other key elements of the adoption methodology are continuing management sponsorship and the identification and resolution of risks.

On the word *process*, the interviewee suggested that it could have a positive or a negative effect on the audience. As a result, the sales person has to be adept at deciding whether to use this word or not. On the positive side she thought that the word *process* could imply quality in the product. One can achieve quality in a product as a result of having a well-defined process that is audited, and that everyone understands and follows. She raised two negative issues that customers perceive with respect to process– one issue is relevant during sales, and one is relevant after sales. During the sale, when people hear the word process, they think of control. However, the interviewee felt that her company's CM tool imposed little overt control (this issue is discussed further in the next interview). After the sale has been made, she believed that the technology transition team needs to be careful about how they use the word process, because developers do not want to be controlled. With an effective adoption team that understands the culture, this should not be a significant issue. However, she felt that was the exception rather than the rule.

Impacts and Insights

Part of the adoption process is the identification and management of potential risks. The interviewee generally found that organizations had about 30 risks. Only about 10 to 25 percent of these risks are related to the product. The others are related to the company in the areas of politics, culture, personnel issues, legacy problems, and the process environment.

The interviewee believed that many of a company's background experiences result in impediments that need to be addressed and resolved before the organization can successfully undertake a change or introduce a new process. Organizations that can not or will not resolve these impediments are negatively affected every time they try to change the current ways of

doing things. The interviewee claimed that "people who don't understand the change process won't succeed," and identified this as a major inhibitor to adoption. Solutions to this issues are:

- introducing incremental change
- keeping models and methodology simple
- setting expectations
  - success criteria
  - requirements
  - metrics
  - start/end of process
- understanding that adoption is about people
- identifying and managing risk
- focusing on the positive

C.5.2 Interview 2

The second person interviewed was the Vice President of Research & Development. He is responsible for developing the internal processes and process support for the development of their CM product.

Business/product characteristics

The R&D group is responsible for designing new product features, issuing patches, performing tests, and producing release tapes and CD-ROMs. However, as an internal responsibility, this group maintains a collection of processes to support the development efforts. Thus in addition to supporting external customers, the group also has an internal support function.

The company is very process-aware because of the nature of the product sold. The interviewee believed that this process awareness has helped them do much more with fewer people. He stated: *We are able to develop more with an average of a third less people. I think this is partly due to our tools and processes.* He especially credited this to the fact that they have a clearly defined development process and that they are a small organization where decisions can be made quickly.

For more details on the CM product that the company sells, see the previous interview summary.

Process maturity

The company has not had an SEI process assessment. However, they are very process-aware as is evident both from the way they operate and from the characteristics of the CM product that they market.

Processes are managed in a flexible manner; they provide a structure for the development activities without being intrusive. The textual support guides one through what operations are needed to perform, for example, the operation of deriving a patch configuration. Metrics are collected informally. The quality manager provides reports on how well the development efforts are going with data often generated by the change management system. Such data reports on problems identified, the quality of modules, etc.

Application Focus

The application in this context is viewed as the CM tool itself. Both macro and micro planning guide the development effort. Macro planning is typically performed using project planning tools such as Mircrosoft Project, while day-to-day operations are controlled by the task management component in the CM tool, which is used in-house to drive development. There is a loose one way feed from task management into project scheduling. Development efforts are driven by the task management portion of tool. The development effort is tightly controlled through the tool - nothing can be done without an approved task. However, there is a "quick task" function the developers can use to bypass the tight control. The quick task automatically creates a task and assigns it to the development team. The developers can do their work but the task still needs to be approved. This was a bottom-up process changed to support the developer.

The planning and management of the development of new releases is fairly loose. Development teams are small. Initially, tasks are assigned at a level that has large granularity (for example, the graphical interface). The development team then comes up with the high-level tasks for the project management tool. For a 9 month project there might be 70 tasks. However, there are hundreds of tasks in the task management system that the developers create. They were not created in advance - tasks are created as they go. Thus a prototyping approach is used.

One issue that we ran into with several interviewees was whether one should let the states of data control the process as opposed to a manager externally controlling the process (the event driven approach). This interviewee stated: *The [CM product] has a data-driven philosophy... [Management] control is way more effective, but it is also less accepted. The big thing is that we stay out of the realm of making people go do this [task] first, then this second, and so on.*

Use of tools and technical development

The CM tool that is marketed by this company is the major software environment used to support development of future versions of the CM system itself. Analysis and/or design CASE

tools are not generally used in the development effort – they are available but not required. Prototyping is required for all enhancements. The process for developing a new feature includes:

- submitting of a new change
- discussing requirements
- writing requirements documents (10 - 50 pages)
- prototyping (may be a fully functional prototype)
- writing a design note for review by architects
- holding a design review meeting
- implementing code
- reviewing documentation

The scope of the feature determines the length of this effort. It can be as little as three weeks or as long as one year.

After a release has been made, a post-mortem is held. At this point, a review of all the processes is made, and processes that have not worked well are evaluated. Thus processes are frequently changed or improved. All processes are stored as versions in the CM tool database. Processes are defined in sufficient, but not excruciating, detail to do the task. The interviewee stated that he wanted processes that "churn" frequently because their own test metrics showed that the majority of the defects are found from internal product use. This approach is found to be more effective than use of inspections or other more formal techniques.

Team characteristics and experiences

The interviewee felt that the biggest advantage that the company had was that *the people hired for development are people who are really into process and want to do process automation. They are tool fanatics. They want to work on this stuff and they want to make better processes.* This common denominator results in a unique situation in which almost every process improvement comes from the bottom up. However, the interviewee felt that these people were still developers and would not go out of their way to do something just for the sake of "process." They would do it if it were unobtrusive and if they could do it well. If something were perceived as being bureaucratic, obtrusive, or constraining, it would not last past the prototype stage.

Engineers and managers receive daily reports from the task management system that identify the outstanding tasks. When they come to work, all developers see which tasks have been assigned to them, which ones are still pending, and which ones are completed. This allows them to rapidly decide what needs to be worked on that day.

Transition and adoption

The company does not need to transition its product internally – it is already institutionalized. However, the company recognizes a significant need to support transitioning of the product to its customers. To this end they have developed strategies to help customers successfully incorporate the technology in their organizations. This is described in some detail in the previous interview summary.

Impacts and Insights

The company is very "process" oriented – they practice what their tool preaches. However, their approach to the use of process is quite low-key in that specific processes are not mandated from above, nor are process controls imposed from above. Generally people who join the company bring a positive process perspective with them, and software developers are involved in defining and improving existing company processes.

The CM tool provides process support through the management of artifact states. Thus developers are free to select what they work on, constrained only by the state of the artifacts associates with that project. Thus overt management control is minimized and freedom of task selection is maximized. In addition to the processes defined by the CM environment, documented manual processes are also extensively used. These are frequently revised as a result of lessons learned from their use in completed projects.

While this company's focus is configuration management technology, and not explicitly process automation, many of their experiences have strong bearing on process automation. In particular, this organization is one of the few we observed that successfully institutionalized a technology with a significant process focus.

C.6 Project F

The person interviewed was a Professor of Information and Operations Management and Researcher at a major university. He is responsible for developing a software tool that supports the engineering of organizational processes throughout their life cycle.

Business/product characteristics

As a research organization, this group's mission is to define a knowledge-based approach to the process life cycle. Development in this area has been ongoing and evolving over an eight-year period. The original focus of this group was pure research. However, over this eight-year period they have developed a wide user base of business clients to test and prototype this approach.

This research team believes that its approach goes beyond the typical process-centered environment by addressing and supporting process environment generation, enactment, and replay. They feel that the strength of their tool lies in its ability to support the full process life cycle. While this tool is primarily targeted to engineering organizations, it can be applied to complex technical domains and to conventional business processes.

Process maturity

Because it is an academic organization with a focus on research, there have not been any in-house process-improvement activities.

Application Focus

The aim of the research is to cover a wide range of process life-cycle activities. As such they are addressing process issues in

- modeling and meta-modeling
- analysis and simulation
- visualization
- automated enactment
- monitoring
- diagnosis and repair

The tools are based on a large process-resource hierarchy, where all objects are descended from (i.e., are sub-classes of) a root resource.

Use of tools and technical development

Support for organizational processes entails more than modeling and the creation of process descriptions and relationships. The goal of the group is thus to provide a tool suite that supports the engineering of organizational processes across the process life cycle. This life-cycle approach is founded on the incremental development, iterative refinement, and ongoing evolution of organizational process descriptions. The life cycle is comprised of the following activities:

Table C-1: Lifecycle Elements of Process Technology

Up Stream	Mid Stream	Down Stream
Meta Modeling	Visualization/Training	Enactment/Instantiation
Model Definition	Prototyping/Training	Monitoring & Measuring
Analysis - Static - Dynamic	Administration - Scheduling - Staff, etc.	Administration - History - Replay
Simulation - Knowledge-based - Discrete Event	Integration - Tool - GUI	Diagnosis and recovery

Table C-1: Lifecycle Elements of Process Technology

Up Stream	Mid Stream	Down Stream
Transformation and optimization	Target Environment Generation	Evolution - Adaptability - Model Management - Model Repository

Meta-modeling provides the ability to address process level interoperability by representing families of processes. The first type of information modeled is sub-tasks to a process (sequential parallel, iterative and conditional). The second type of information in a process is the resource requirements, that include people, input, tools and output. The third type of information is the resource instances that are assigned to the resource in order to match the resource requirements. The last set of information is for scheduling requirements.

The group is investigating the need to deal with unexpected real-time failure during automated process enactment. The interviewee expressed the belief that this is a real limitation with most currently available enaction tools and that solutions be actively sought.

Team characteristics and experiences

The team is primarily composed of university staff and graduate students.The development of the process tools is a combined effort between academic, business, and government agencies.

Transition and adoption

The group provides technology that helps process adopters do their jobs. Such support is in the areas of

- simulating walk-throughs of processes
- providing graphical and animated representations of process
- capturing and evolving process definitions

Impacts and Insights

The interviewee felt that the following insights resulted from his experience with the practical application of his research program:

- Graphic visualization
  - is an important element in communicating process issues to managers
  - supports intuitive analysis and discovery
  - encourages acceptability of process
  - frequently impresses audiences

- Wide area heterogeneous networks, particularly using the Internet, are becoming of increasing interest.
- Most "as-is" processes are vague and not clearly understood.
- Baselining of "as-is" processes is not often performed prior to installing revised processes.
- Prototyping and walkthroughs are effective means of generating end-user feedback.
- Simulation is a high pay-off technology that is insufficiently used.

C.7 Project G

Interviewees

A technical lead for a demonstration project involving new processes, tools, and method technologies was interviewed. The interviewee mainly helped to develop the process that was automated, but he also had some minor involvement in advising the demonstration project in use of the new processes and tools.

Business/product characteristics

The organization in which the demonstration project was run was a large DoD software support group, primarily involved in the maintenance of weapons systems.

A large government technology program provided funding to the organization to create and run the demonstration project. In turn, the organization was asked to provide feedback regarding the technology developed. The government technology program expected to gain insight into the benefits and drawbacks of the technology, and into approaches for technology transition. On the other hand, the receiving organization would be able to determine the effectiveness of the technology and the ability of its staff to use the technology to perform software development/maintenance activities. Since funding for the demonstration project was provided by the government program, the risk to the organization was significantly reduced.

Process maturity

It is not clear whether the organization in which the demonstration project was run had been formally assessed, but the interviewee described it as being a "classic" CMM level 1 organization (ad hoc). The organization's management was sold on the idea of process improvement, however, since word had come down from higher levels of organization that all subsidiary organizations must reach CMM level 3 in order to continue to exist.

The technology brought into the organization for demonstration centered around the methods and tools used to support the Cleanroom approach to software engineering. Previous to the

demonstration effort, the organization's personnel had no experience with Cleanroom. Thus, the process that was executed was completely new.

The interviewee suggested, however, that a number of factors worked to rapidly improve the process capability of the team involved in the demonstration project. An employee of a vendor quite familiar with Cleanroom was used to define the initial process. The resulting process was then refined by the interviewee while on temporary placement at the SEI. As a result of this strong process focus, the interviewee felt that the resulting process was extremely well-defined and structured, and could be readily adopted even within a less process-mature organization.

The personnel who made up the demonstration project were not initially process-aware, with the exception of the team lead, who had some SEI training. Since that time, the group has had more SEI training. The degree to which they are now involved in process improvement is demonstrated by their effort to map Cleanroom to the CMM Key Process Areas (KPAs), and to use Cleanroom-like techniques to address missing KPAs.

Application focus

The demonstration project involved reengineering the software of a small weapons system. The target system consisted of approximately 30K SLOC of Ada.

Use of tools and technology

The pilot project brought in both the Cleanroom approach and associated process-driven project management tools. The initial process automation capability (Version 1) involved only a small number of tools. These tools supported the defining of roles for team members, and the assigning of tasks to team members. While it was quite rudimentary and inflexible, Version 1 worked well in saving process state and generating reports from the underlying Oracle database. However, modifications to the process were awkward to deal with, since Version 1 was written in shell script. This initial tool-set did not invoke external tools; rather, it focused on assigning tasks to individuals.

The second version of the automated capability (Version 2) attempted to fix problems with the initial version. Version 2 was more dynamic and allowed easier modification to tasks (modifications could be done on the fly). Version 2 was built using a high-level process

automation tool developed for the government technology effort. This high-level tool (here called HLT) was itself built on top of the commercial ProcessWeaver process automation tool.

It was felt by the interviewee that ProcessWeaver[1] was difficult to use and somewhat inflexible. The interviewee stated that the version(s) of ProcessWeaver used also had reliability problems and problems with graceful recovery from power loss. Each time the demonstration project needed to bring the system back up after a problem, they needed an expert to put the tool state back in order. It was also felt that the price for ProcessWeaver was too high at the time (the price has subsequently been lowered).

HLT was developed out of the recognition that there are basic themes in process automation that are constantly repeated (e.g., release document for review, review document, etc.). These basic themes can be considered to represent generic tasks. HLT functioned by instantiating generic Petri nets reflecting these themes, that could then be entered into ProcessWeaver for enactment.

Team characteristics and experiences

Previous to the demonstration effort, the 12 people involved in the effort were primarily involved in subcontractor management. The group was specially assembled for the pilot project from contract management/supervisory personnel and it did not have a track record for software development/maintenance.

The 12 people working on the demonstration project came from 4 different departments, and initially were not all located in same building. They were all civilians with a fair level of experience, but were not senior engineers. Their ages were primarily late 20s and early 30s. Members of the group had a computer science background, but were not development experts.

Members of the team spent approximately 25 percent of their time working on the demonstration project. The rest of the their time was spent working on their normal duties. This was considered to be a disadvantage because team members were not totally immersed in rigorous Cleanroom process.

Transition and adoption

Originally, the group had an attitude of "why us?" It expected failure because of a history of poor communication, and the fact that four different managers were involved. In particular, the individuals involved in the certification of components did not understand what was expected

1. Note that the process automation tool ProcessWeaver from Cap Gemini has been used quite extensively by the projects interviewed for this study. Because of this broad experience, some of the weaknesses of this tool are identified. This does not imply that ProcessWeaver is in any way inferior to other comparable tools. In fact it is to ProcessWeavers's credit that it was chosen as extensively as it was by the organizations interviewed.

and lacked management support for their activities. The interviewee considered certification to be the most intellectually challenging part of the Cleanroom process.

Members of the demonstration project team (the end-users of the automated capability, not those developing or championing it) were not involved in defining the original process automation. The parameters of this automation came from the Cleanroom methodology, modified by the interviewee when he was at the SEI. However, when the end users became expert at Cleanroom and the toolset, they made numerous suggestions for improvement.

There was little or no direct contact between process developers and the team using the automation. Other consultants (employees of the same firm as the interviewee) served as middlemen. The interviewee did not expound on any problems this might have caused, because his primary role was in the area of process definition.

Training was provided both in process and tools. However, initial training was provided four to five months before actually starting the job. Personnel had to be retrained, and lots of "hand-holding" was required. The interviewee stated that Cleanroom is quite demanding, so that appropriate and timely training was critical.

Implementation planning for the process automation capability and the Cleanroom methodology was performed primarily by the contractor representing the government technology program. The local organization provided some support. All told, the contractor spent about one year working with the demonstration project. However, within four to six months, members of the demonstration project had gained intellectual control over the process. Even then, on-site support was still critical for situations not documented in the manuals.

The approach taken for bringing in the process automation was certainly not incremental in that both a new process and a new process environment were brought in at the same time. However, incremental release of tool versions did help. In addition, the interviewee suggested that the "part time" nature of the activity for the 12 participants may have actually reduced some of the pressures of a "big bang" approach, since they could fall back on other, more familiar processes for a large part of their workday.

There was no on-site support for the tool, and this led to long delays when the system went down. There were also communication problems between pilot personnel and the contractor supporting the government technology effort, that delayed response to critical problems.

The contractor initiated a number of special adoption support efforts. Among these were

- setting up a special project work room to encourage personnel to come together in one place and develop team spirit. The interviewee felt that such structural changes are important, because the existing organization had to be broken up in order to encourage group cohesion.
- accommodating individuals where possible, but using peer pressure to bring people in line (for example, emphasizing "team" errors). The interviewee

presented the example of an individual who was allowed to do things his own way. However, when the individual's work was sent for certification, it did not match the expectations, so the team was flagged for many "team errors." This led to a great deal of peer pressure to conform.

- providing methodological support for Cleanroom and tool use. The interviewee noted that some of the problems experienced in the Certification activity could be traced to lack of contractor expertise in the Certification process.
- moving Certification personnel into the same building, and setting up special meetings to bring this group (two to three people) in line.
- training on the Cleanroom methodology and the process automation tools, tailored for the demonstration project.
- on-site coaching in the Cleanroom process.

Impacts and insights

Tool Related Impacts and Insights

The interviewee believes that the project was probably not large enough to be a good demonstration of process automation technology. However, he believes that such technology can scale well if it is sufficiently flexible.

General lessons learned about process automation tools included

- It is essential to provide tool support that maintains flexibility in the process.
- Process automation tools must be robust before they are transitioned to end users.
- The use of a process automation tool does not reduce the need for good process definition and management.
- Communication was enhanced as a result of both the Cleanroom process and process automation capability.

The demonstration project experiences with the overall effort were mixed. While the interviewee felt that the process automation capability was essential for introducing Cleanroom and supporting Cleanroom adoption, the tools eventually proved to be too inflexible and defect-laden to support a working project. Both HLT and Process Weaver were eventually abandoned by the pilot project because of inflexibility of, and defects in the toolset. The Cleanroom methodology is still in place.

A major configuration problem involved keeping consistent different versions of tools (Process Weaver and HLT). The organization developing HLT did not respond well to requests from the contractor, and had little interaction with the actual project. HLT was also poorly constructed, had many defects, and required frequent patches.

The process implemented in HLT was also too rigid and confining. For example, a member of the demonstration project needed supervisor (manager/lead engineer) permission to browse

files in the program library. Unfortunately, the developer normally has little a priori idea of what she/he must review. Members of the demonstration team circumvented this particular problem by giving all members of the team the role of supervisor. Unfortunately, this had the side effect of giving them all the same privileges afforded the manager. The HLT builders could not be convinced to loosen these types of constraints.

Interestingly, as more updates were made to HLT, the capabilities of Process Weaver were used less frequently, primarily because the new features built into HLT were similar to Process Weaver features. In addition, both the demonstration team and the contractor were concerned about the high cost and inflexibility of Process Weaver.

The interviewee suggested that automation is the key to helping people adopt complex, demanding processes like Cleanroom. Even though the demonstration project group has moved away from the process automation capability, he feels that Cleanroom itself would not have been accepted without the process support.

The interviewee further suggested that contractual incentives must be provided for people to use new process automation technologies. Without such incentives, the interviewee suggested that people are not likely to adopt process automation capabilities.

Many of these lessons have been incorporated into a new process automation toolset. Unfortunately, the demonstration project has abandoned the technology and the contractor is making no real use of the work done for the government technology effort. There are also no known contractor projects using Cleanroom, in part because Cleanroom advocates have been somewhat heavy-handed and dogmatic in preaching the approach. The interviewee suggested that pieces of the Cleanroom process and automation can be brought in to address specific situations, without any complete commitment to the approach.

Process Related Impacts and Insights

There is evidence that some end users initially found the Cleanroom process to be overly confining. This evidence involves anecdotal stories of project members who preferred to do things differently. However, the general impression left by the interviewee is that the Cleanroom process is reasonable for automation, primarily because of its well-defined and controlled nature. The interviewee did suggest that tightly-defined processes such as Cleanroom should be tailored for the needs of the particular organization.

As a result of the demonstration effort and the Cleanroom process in particular, members of the demonstration team now have a much greater focus on metrics. Prior to the demonstration, they did little or no metric tracking of projects. As a result, it was difficult to acquire baseline data for the organization, but a "best guess" identifies productivity in the range of 120 SLOC per person-month. Now, members of the demonstration project are consistently tracking productivity, error rates, and other project characteristics. Team

members have seen a 400 percent increase in productivity, with error rates below 1 per 1000 SLOC.

Based on the metric data the demonstration team is now gathering, they are bidding on and winning new jobs. One job was won primarily because the team had real numbers that showed high productivity and low error rates. However, their initial bid was rejected because it was "too low." They then increased the bid and won the contract.

The 12 people in the pilot have decided that for all future re-engineering work, they will use the Cleanroom approach. There is a general fear that Cleanroom costs will be somewhat high for minor maintenance work, but for re-architecting the system, costs seem reasonable. It has been decided by the team that if more than 25 percent of the code in a system is to change, then the Cleanroom approach will be used.

Organizational and Team Impacts and Insights

In general, the group is very enthused about Cleanroom, but are less so about the automation. They exhibit a strong esprit de corps, and are proud of their productivity and quality improvements. Their managers have stated that if the only improvement had been in morale, then the project would have been a success. However, even greater successes are evident.

Members of the demonstration project have expressed relief and improved job satisfaction because they have a well-structured way of doing work. They no longer are required to "make it up as they go," and can focus on the job. However, there is a fear that they may revert back to the old way of doing things now that there is less of a spotlight. There is also some underlying fear that they will be overwhelmed by the prevalent culture. This fear is probably justified since only 25 percent of their effort is involved using the new approach, and they represent a select group within a larger, more chaotic organization. Team members also express concern that the demonstration team will be broken up.

The demonstration project personnel are taking Cleanroom back to their "home" groups. However, there has been no seeding of other groups (outside of the home groups) with these experts.

C.8 Project H

Business/product characteristics

This software lab was formed seven to eight years ago when the parent division was still part of a major aerospace company. It was formed to reduce the need for projects to reinvent development approaches. Specifically it was tasked to develop and transition standards, methods, and tools. With respect to tools, they developed a technology with which to implement process automation.

Process maturity

The automation tool was viewed as a tool supporting CMM Levels 2 (the Repeatable Level) and 3 (the Defined Level) through its process and product management capabilities. Regarding Level 2, many tool features support project management functions and make them repeatable. Regarding Level 3, by embodying a standard process within the tool, the tool provided a foundation for an organizational standard software process.

Application Focus

The project was initiated in 1989. The environment supported the software development activities in systems development. Specifically, it addressed software project management, software requirements management, software design, software code and unit test, and software integration testing. The objective for the tool was to enhance product quality and developer productivity. To achieve the objective, the tool integrated data and task flow as well as development and presentation tools. The environment also provided support for project management.

The environment was originally conceived to support DoD standard 2167 and Ada. Due to this, the tool developers did not identify a need to allow for flexibility and customization. However, this original concept changed during the course of the environment's development. Subsequently, the design was modified to include features permitting tailorability of the process. In addition, the environment design was role-based; developers only saw developer steps and managers only saw manager steps.

Use of tools and technical development

The tool is primarily composed of COTS components and is based on an open systems architecture using UNIX, X Windows, and Motif. The COTS components support project management, software analysis and design, code development, integration and test, and user support. In addition, the environment used a common object repository to manage data from the tools and to store organization-specific data such as standards and procedures and templates.

From the end-users' perspective, the tool had much to offer in theory, but the developers did not anticipate the reaction of the end users. The concept did not reflect the real needs of those planning and managing projects. Rather, it tried to force an orderly process on people who did not follow an orderly process.

The original idea for the automation tool grew out of a previous tool development effort. This latter tool focused on using PERT chart techniques to produce a three-point estimation that helped with planning, and produced a probability distribution on project completion time. However, only a few managers could cope with the tool because firstly, it forced them to think at too low a level of detail and secondly, the user interface and operation of the tool were not intuitive.

The automation tool required a similar level of detail as the previous tool. Too much detail was required too far in advance to use the environment for planning. Managers found it was not

helpful and took too long to enter the data. The process envisioned by the tool did not match the way managers worked. For instance, the tool did not readily support replanning or developing project-level process model.

The organization used 2167-based processes, and these were embedded in the tool. These cradle-to-grave life-cycle processes were not rigidly followed but were adapted for use in the tool. Once embedded in the tool, they were not easily tailorable, and this reduced its attraction and usefulness to the software projects. Project members had to go to tool experts and developers to perform process tailoring within the environment in order to add or delete process steps.

Team characteristics and experiences

The tool was developed by members of the software lab with relatively minor involvement of personnel from outside. The development of the environment became a major project for the lab, that at one time employed approximately forty people, but is down to fewer than ten.

Transition and adoption

The approach for development and transition was to create an entire software engineering environment and provide it to projects. The idea of releasing capabilities in phases was rejected because it was thought that senior managers would view that as too little functionality for the money. So, the tool effort attempted to develop an environment supporting most of the software life-cycle. The tool project developed into a demonstration project that received varying levels of interest from different company divisions. However, no pilots were ever conducted. A commercial vendor, upon whose infrastructure technology the tool was implemented, will finish development and commercialization.

As the automation-tool developers analyzed existing tool use within one organization, they found that the software engineers used some tools whose design was unclear. These tools had been handed down and the original authors had moved on. No one knew how to maintain them and hence, people were afraid to tamper with them. This was viewed as a risky foundation upon which to build automated processes. Alternatively, at another site, the tool developers found good process definitions and use of them. However, the existing environment was primarily manual and offered little tool use that could serve as a foundation.

There was a feeling among the tool developers that the user organizations were not ready for an integrated environment. To some extent, the project encountered the "not invented here" syndrome. Furthermore, while they had good support from senior management, support was lacking from the middle management ranks.

At the project level, support was also lacking. In particular, no funding was provided to do the technology insertion – developers were not funded to learn and use the new tool. Nor was any

commitment made to long-term use of the environment. For software engineers on a project, there was no motivation to learn to use it on their own.

Some of the observations by the developers were also echoed by the end users. Drawbacks perceived by the end users included

- There was no training on tools that were part of the environment.
- There was no real piloting of tools and the environment.
- The tool development environment was older than the production environment.
- There was no flexibility in using the tool.

Impacts and Insights

The developers identified organizational characteristics associated with successful technology change and introduction. The conditions they believed were necessary for success included

- a gradual approach of laying a foundation and building upon it through incremental introductions of capability
- ensuring that the organization has processes, methodologies, technologies, and tools in place prior to introducing an environment

Upon re-examination, they felt many of the necessary conditions were lacking in the company divisions that could have served as demonstration sites.

Some of the lessons learned by the developers include

- There are certain foundational characteristics necessary for the successful introduction of an integrated software engineering environment. These characteristics were not found among the potential pilot sites.
- Mid-level and project-level management support are necessary for success.
- Introducing the environment in phases rather than all at once will reduce "not-invented-here resistance."
- Introduce tools to support small teams rather than the whole project.
- Develop tools and an environment that helps the teams, rather than solely providing data to others.
- Companies that do not spend much in computing will not spend much for an automation tool.

C.9 Project I

Business/product characteristics

We interviewed members of a team charged with maintaining a French money exchange network for bankers. The team was responsible for keeping the money exchange network in working order and for analyzing and correcting anomalies found by bankers.

The team consisted of between 10 to 15 persons, with a manger and two sub-managers, each with a staff of four to five. The offices of the team members were located in close proximity to each other. In spite of the existing structure, the team did not function in a strictly hierarchical manner.

Process maturity

The team supports three main processes: one for small problems, a second for more serious anomalies, and a third process that is invoked if one of the two other processes is operating abnormally. Usually, activities go smoothly – the third process is scarcely used. While the processes were not ad hoc prior to the implementation of the process automation system, the advent of the automation system has required that the team spend time identifying a more formal definition.

The unit does not make use of a true CM system. However, they do track a number of indicators of the state of the system. These indicators (metrics) included: the number of anomalies reported, the delay for each step in the various processes, and the effort required to correct each problem. Delays are very important to the unit, because the contract contains a clause specifying that anomalies have to be resolved within a certain time period, or contract penalties will be applied.

Application focus

The three processes described previously, representing almost all of the team's activities, have been automated. The actual processes are not complex, and the largest process can be displayed on four monitors. The processes are primarily composed of sequential steps. As designed, each process can involve up to five individuals, but in practice, two to three persons are normally involved.

The expected (and realized) benefits of the process-centered environment include automatic alerting of personnel when the solution to a problem is delayed (or is taking longer than expected), and a straightforward mechanism to identify similar problems and their solutions by querying a problem database.

Use of tools and technical development

In preparation for the process automation tool, paper and pencil techniques were used to specify the processes for encoding. Personnel from the group that had developed the tool (the tool was developed in house) assisted in the process definition and encoding. Experts in the tool suggested that the encoded processes should be simplified, but to this point only slight adaptations of the processes have been necessary.

The infrastructure supporting for the process-centered environment consists of a ten-PC LAN with a UNIX server. A central component of the environment is a database that is automatically updated to reflect the current state of activity of each known problem. Those responsible for administering the infrastructure expressed reservations about the reliability and performance of the automated environment due to the complexity of the architecture needed by the database. Initially, there were frequent system failures, but subsequently the rate of failures has decreased, and now about five re-boots per month are required. The cause of the initial failures was faulty configuration of the PCs. However, end users now appear satisfied with the system.

Team characteristics and experiences

The manager of team has been with the company since 1988, and was quite familiar with the project and work processes prior to the automation effort. The team members are engineers and have an average of one year of experience with the company. They were generally in favor of the transition to the automated process because of the benefits provided by access to historical problem data stored in the database. One team member (affiliated with a trade union) initially felt that the tool was an intrusive "Big Brother." However, this problem was resolved by making the data in the tool open to everyone and ensuring that tool data was not used to control people.

PCE end users were not specifically trained to the automated process. According to the manager, training was not necessary because the individual engineers understood their particular

job, and did not need to understand the global process. The tool was put in place primarily to support the engineers in the jobs they were already accomplishing manually.

Transition and adoption

The decision to install the tool was made by the manager (with his manager's approval). The tool was viewed as a way to assist in the management of the 15-person team. Introduction of the tool involved an abrupt cut over to the new system. This approach was necessitated because of modifications in the database interface and platform. Before the advent of the PCE, end users (engineers) were required to input data manually into a PC-resident database. With the PCE, data is automatically entered into the server-based database.

This abrupt introduction was made less onerous by the fact that end users were happy to be migrating from a time-shared mainframe to individual PCs. The manager also introduced a system for suggestions and complaints. For the most part, suggestions and complaints involved only ergonomic problems.

Due to technical problems, the implementation phase was quite long and difficult. Since the very beginning (eight months previous), a person was assigned full time to deal with the administration of the system. However, the early problems have now been ironed out and the administration job is viewed as less necessary.

Impacts and insights

The tool has had a number of positive effects, including

- The team was obliged to develop a more formal definition of the three processes.
- Tracking of personnel assignments and problems is simplified; they now can quickly tell who is working on what.
- Individuals are better aware of their work assignments.
- The automatic updating of the database provides straightforward access to historical information concerning problems and fixes.
- The process can be verified from data saved by the tool, thus ensuring consistency and quality of the process.

The most significant technical problem involved the development of the link between the process automation capability and the database. This link required special development work by the process automation team, and integration of the two components was troublesome. The link is now functioning correctly, but the implementation is not technically satisfactory, and the developers would likely implement the link differently if they were given an opportunity to start again.

The team of end users experienced few adoption problems. They are happy to have quick and open access to problem information, and are pleased with the PC working environment. They believe that the process automation improves the quality of their work. They do not feel as if the system is permanently monitoring them. They feel that they have avoided loss of control of their work to the automation capability (becoming "robots"). However, they do suggest that concerns about loss of control can become significant if not addressed while designing the process.

At this point, the process automation capability is working well, resulting in transfer of staff to new developments. The team manager has become known as a PCE tool expert, and is involved in several demonstrations of tool capabilities both within and outside of the company.

In general, the interviewees believe that the tool will be most useful for documentation management and for coordinating groups of more than five people involved with complex processes. Use of the tool does imply that each user has access to a PC, and that may not be possible in some organizations.

C.10 Project J

Business/product characteristics

Employees of a large aerospace company were interviewed. The specific unit interviewed develops and maintains real-time, embedded hardware and software that records flight data. The data is analyzed to facilitate aircraft maintenance. Clients of the unit are both internal and external; other units within the company rely on software and hardware produced by the unit, and the final product is sold to an aircraft manufacturer.

The entire unit consists of ten teams, but only two are involved in the PCE experiment. The first of these two teams (specification and quality control), has one primary user of the PCE; the second team (development and validation) has five PCE users. Thus, the total number of PCE users is small. In each case, PCE users are team managers: the company does not support the use of the PCE by individual contributors. However, the managers using the PCE work in units with hundreds of employees. The large size of the company results in a somewhat bureaucratic organization. There are a number of very formal and official paths to be followed

Process maturity

In general, processes within the aerospace domain are very precise and well-described. Commitment to process improvement within the two involved teams is strong. The development and validation team earned ISO 9000 certification at the beginning of 1995, while the specification and quality control team is also expected to achieve ISO 9000.

In spite of deadline pressures, work usually goes smoothly. An interactive approach to development is used. Team managers are more or less free to choose their management approaches and tools.

Application focus

The aim of the PCE effort is to automate the main processes of the specification /quality control and development/validation teams, and then link the two processes together. The processes involve the management of software evolution, including both technical and administrative communication.

Project goals include reducing the amount of time it takes to complete work, increasing the dependability of the process (and ultimately the systems produced), and providing a tracking mechanism. The PCE is also expected to ease communication by providing a consistent view of the state of the process and by insuring that processes run correctly. Another benefit is that the PCE is expected to speed and simplify quality certification. Finally, the PCE is expected to allow the company to change and improve processes more quickly.

The experimental PCE efforts will be evaluated after a year. If results are positive, support will continue for the current PCE efforts, and new PCE efforts will be initiated.

The organization would like to use the PCE to identify and measure process delays, and record the number of cycles needed before a satisfactory result is obtained (e.g., the number of iterations before a validated document is obtained).

Use of tools and technical development

A client/server architecture combining HP and Sun servers (with a bridge between) connected to PC and VAX/VMS clients (respectively) has been developed. A number of tools are being integrated into this environment, including WordPerfect, Oracle, and a CM tool. Integration of the tools has caused some problems and has led to a number of specialized development efforts.

Another problem involves poor reliability of one of the servers: The Sun server experienced six breakdowns over a three month period. While the number of breakdowns is not in itself too high, some of these breakdowns lasted three days. Such lengthy breakdowns can stop an entire process when the process is supported by a PCE; this situation was unacceptable. However, the causes of the failures were found to be hardware problems, and these have now been fixed.

As a first step in the development of a PCE, the teams are building process specifications that describe current processes; as a second step, they will optimize the processes. Usually, before encoding processes in the process automation tool, the processes are simplified (e.g.,the teams ignore all the possible error cases).

Despite the fact that the process automation work was in the process specification phase at the time of the interview, the use of an iterative approach has already allowed end users to test early versions of the PCE. End users were very cooperative and made a number of suggestions for upgrading the processes. This early use also pointed out that there were some gaps

between the theoretical processes encoded in the automation and those actually carried out by end users.

The organization has not considered alternate process automation tools, because of the complexity and time-consuming nature of the selection process. The interviewees suggested that tests of process automation tools should last several months and should be conducted on real cases. The long test period is mandatory when dealing with a type of tool that is not yet well accepted by end users; trying any such tool involves waiting for a change in the attitudes of these users.

Team characteristics and experiences

The specification of the processes has been driven by user (in this case, manager) interviews; thus end users are involved in process definition and even validation. The average user has about six years of seniority in the company. Except for one individual, all end users are computer engineers or are at least well-acquainted with information technology.

The two interviewees in charge of process definition are trainees finishing their university degrees. The interviewees have taken courses on process specification, but are not experts at the work of the organization; they are only at the company for a six-month period. However, they were trained in the use of the tool at the very beginning of the project. The timing of this training proved to be too early, because the interviewees could not use the tool immediately after this the training session. Another training session is foreseen.

Almost all end users are in favor of the project; for them it is an experiment that will provide them with more powerful computers (VAX stations). The single reticent user was concerned about the possible rigidity of automated processes. However, he is open-minded about the activity and in fact has made the greatest number of constructive comments.

Transition and adoption

The introduction of the PCE is an experiment initiated by the company in partnership with the tool provider and the European Community. The support for the experiment was driven by a strong need to document their procedures and to track their processes, as well as the desire to achieve ISO certification. The experiment is expected to last one year; at the time of the interview a third of that time had passed.

The interviewees feel that they have already have achieved positive results from the process specification activity. However, they cannot really say what they have gained from using the process automation tool and PCE, because they are still in a definition phase and they have experienced integration problems.

End users have not rejected the tool and are in favor of the project. This may be due to the experimental nature of the project and because the company has taken time to communicate the nature of the project to employees. The positive feeling about the project continues because end users have already obtained good results with the process definition phase. To

introduce the PCE, the developers chose an iterative method with a short cycle time: the process is specified, then validated conceptually by end users and managers, and finally implemented and tested on real cases. The implementers believe the tool is well-adapted to this iterative strategy.

Impacts and insights

Some changes have already been made to the process based on use of the process specification activity and early feedback from the PCE; some activities were deleted and others modified. No changes in staffing have occurred. However, if they were to start the experiment over again, they would introduce the tool later than they did to increase the time between the first phase of process specification and the first phase of implementation.

The biggest technical difficulties have involved the integration of the different tools and non-optimal response times when using the PCE. Identifying the cause of poor response times has been somewhat difficult due to the complexity of the architecture. They also noted two other potential technical difficulties: difficulties in maintenance of the system (they feel maintenance must be very good) and in the identification of good indicators for the state of each process.

Because they are still in an integration phase, the team has reached no firm opinion on the value of the automated processes, but they can already state that they received some benefit from specifying their processes. They believe that the PCE should be valuable when there are many administrative users; in this case, the PCE can simplify administrative exchanges by reducing paperwork and the number of phone calls. For instance, the PCE should support very efficient document management.

In the case of this organization, the PCE is expected to have a greater effect on validation activities than on development activities. The team expects that other units will embrace the process specification activities supported by the PCE experiment. If the first tests of the actual PCE provide positive results, other units will probably also adopt the PCE. However, they also mentioned that units will carefully check the price of the PCE solution, since, in most cases, it implies a PC for each user.

C.11 Project K

Business/product characteristics

This unit attempts to help people using new technologies, particularly in the area of process support. The larger organization deals primarily with space-related projects, such as satellites and the European space shuttle. These projects are of a critical nature, with strict requirements on such system characteristics as reliability and quality. The management approach

reflects the strict requirements of the systems, with a strong process emphasis for critical areas. However, management also allows significant freedom in non-critical areas.

Most of the unit's clients are internal, but they have also external clients in other European projects. The unit is small, composed of only two persons plus the manager, but they can obtain additional assistance for special needs or particular domains (e.g., human factors or ergonomic problems).

Process maturity

The organization is involved in process improvement efforts. The interviewee expressed the belief that some units were using the SEI CMM (although there was some doubt expressed here). The interviewee also declared that the organization's processes are very well defined and did not really need to be evaluated.

Adhering to deadlines is the organization's main problem. Otherwise, work usually proceeds smoothly. They use a CM tool in the automated process (see below), but not for project management. To manage their projects, they rarely use metrics: They use metrics to have an instantaneous image of projects but they do not use them for statistical purposes. However, in the automated process they keep track of the time spent on the different parts of the process.

The organization has determined that it will not use metric data to manage projects: their feeling is that metrics have a "big brother" aspect that they want to avoid and also that metrics are generally too difficult to interpret.

Application focus

The process being automated involves correction of anomalies in real-time and embedded software. This process was chosen because it was of average complexity, highly repetitive, and was well-specified prior to the start of the automation effort.

Use of tools and technical development

Over the previous two years, the unit had accumulated experience using the process automation tool to demonstrate, simulate, and describe organizational processes. However, the tool was not used to automate an actual process. The current effort represents an experiment initiated to determine whether the tool could be used to provide process automation support for "real" activities.

Initially, the managers of the unit were a bit skeptical about whether the tool could provide the required level of process support. However, they did approve a nine-month-experiment implementing a PCE based on the automation tool; they are now in the middle of this experiment. To this point, results have been positive.

The interviewee stated that in the space system domain, jobs are very well-described. However, he also pointed out that, despite strong job descriptions, implementing a process for

automation always exposes insufficiencies in the existing documentation. The interviewee attributed this to the differing nature of job descriptions and process specifications.

During the implementation of the PCE, the target process changed considerably because of modifications in the business. For example the process had to be modified to reflect a change from reliance on external suppliers to the use of internal sources for parts. The process has also been simplified, but these modifications were done without major consequences to the automation work.

The activity being automated involves five different kinds of jobs (roles), but the team performing the activity can consist of fewer than five people. The personnel on the team may differ between instantiations of the activity.

The initial process description was built by the members of the tools and methods unit. They started from company job descriptions and then interviewed end users. End users had little chance to modify the process.

End-users were well-acquainted with information technology, and were using a number of different types of computers (mainframes, Macintosh, PCs), as well as a range of different types of tools (editors, compilers, Framemaker, a CM tool,...). The process definition method supported by the automation tool was judged to be unclear and difficult to understand. Instead, a locally-developed method was used.

The automation technology used in the experiment was selected both because of the organization's previous experiences with the tool, and because of their close relationship with the supplier. Only Framemaker and the CM tool were integrated into the PCE; use of these tools was mandatory within the organization. Other tools were not integrated, since end users were allowed to select which specific tool they wished to use. Integration with Framemaker was accomplished without too many problems, but integration with the CM tool was more troublesome. These integration problems were attributed to the fact that the CM tool is an "old" tool, and therefore not as open as newer tools.

Expected benefits of the PCE include greater process maturity, along with an enhanced ability to evolve and improve the process.

Team characteristics and experiences

The process automation team includes three members: a manager and two inexperienced engineers. The end-user team is composed of experienced technicians and engineers (around 5 years of experience per team member). End users were involved in process specification activities, but their ability to modify the process was limited. However, there was, and is, good participation by end users in the project. Both the end-user and process automation teams have received training in the tool; a three day course for end users, with a longer training course for members of the tools and method unit, who are acting as administrators.

Transition and adoption

The Tools and Methods Unit is responsible for selecting and disseminating new technology to the other units. However, this unit has no power to control the technologies employed by other units. Tools and methods may propose new technologies to managers of other units, but the managers of those units can select whether or not to proceed.

The Tools and Methods Unit has employed an incremental approach to introducing the process automation tool and the PCE: This approach consists of demonstrating the capabilities of the tool; formally specifying the process used by the interested unit; requesting validation of the process; enacting the process and integrating the tools; and soliciting user input to refine the interface. The Tools and Methods Unit will repeat these steps as necessary.

The Tools and Methods Unit remains responsible for the maintenance of the PCE (including both the software and process descriptions). This arrangement will continue for the duration of the test and possibly longer, since modifications to the enacted process are too difficult for end users. If at the end of the test a decision is made to continue use of the PCE, the manager of the end-user unit will eventually take over administration responsibilities.

Impacts and insights

The organization foresees two benefits from the process automation experiment: first, the act of developing a concrete process specification is itself an improvement of the process, since the specified process can be verified and is open to analysis of problem areas and improvements; second, the use of a process tool reduces the time involved in some tasks, thus providing the opportunity to focus on more important tasks. This second benefit was unexpected: Their intention was to automate the process, and only after using the automated process did they realize that they had automated part of the end users' tasks.

If they were to start again, they would spend more time on process definition prior to implementation of the automated environment. Their integration activities took longer than expected because they were forced to implement a lot of unforeseen and rare branches of their process; some of this complexity could have been avoided with a longer preliminary phase of process description. Due to these unforeseen detours and because of their difficulties with the CM tool, tool integration proved to be the most difficult issue.

They did not experience any problems with end-user acceptance of the technology. The automated process capability has been very well accepted in the unit for which it was designed. However, during initial presentations to other units, unit managers expressed concerns. It is felt that these concerns were successfully addressed by the manager of the unit involved in the initial PCE test.

The interviewee believes that automated process support can be most effectively applied to repetitive processes, to evolving processes, and also to complex and rarely used process. The benefit in these latter cases should come from the fact that end users are relieved of the burden of understanding or remembering the process; the PCE tool can guide them.

Interestingly, in the interviewee's opinion, the PCE tool is appropriate for managing anomalies and other very consistent processes, but not for software development activities because (at least in his company) these processes vary too greatly from one project to another.

C.12 Project L

Interviewee

The manager of a team involved in the reverse engineering portions of a large management information system (MIS) was interviewed. The project was serving as a pilot of a recently procured software development/maintenance environment with a process automation component. During the interview, the manager involved other personnel to answer specific questions and provide demonstrations.

Business/product characteristics

The wider organization maintains software for military management information systems. It includes a mix of civilian employees, contract personnel, and military personnel. According to the interviewee, there is a wide range of development approaches and skill levels represented within the organization.

The specific pilot project involves reverse engineering parts of a large MIS that supports maintenance activities at military depots. This is being accomplished by building system models for three components of the system. The work also involves generating a high level design for the system, including Ada interfaces. The models and interfaces are intended to be used for the re-implementation of the system in Ada, although the re-implementation activity is outside the scope of the immediate project.

Two of the subsystems being reverse engineered use different commercial databases with different versions of SQL for queries. The third subsystem uses a proprietary database and query language. All of the subsystems are poorly documented.

The project has adopted an interesting approach to managing customer relationships. For each customer (those with a major interest in the success of the project), management plans identifying a successful outcome were developed. Most relevant to this study are the recognition that stakeholders in the pilot project include those interested in the success of the automated environment, and the formalization of plans for those stakeholders.

Process maturity

The larger job responsibilities of the manager include overseeing the implementation of new processes, technologies, methods and tools. One current activity involves the creation of a technical manual defining the software engineering process for the organization. The history

of process assessment and process improvement activities within the wider organization is unknown.

To date, the pilot project has not completed an SEI software process assessment. However, the interviewee feels that the project currently falls within CMM Level 2, and with a few necessary improvements could quickly become CMM Level 3. This interviewee was reinforced in this belief during informal conversations with SEI personnel.[2]

Application focus

The environment that the project is piloting uses the In Concert process automation tool, and incorporates a complex process model that contains over 700 separate activities. The interviewee indicated that the process model delivered with the environment was overly complex for the activities of the project, so a custom process model was created.

The custom process for the pilot reverse engineering effort was developed by the interviewee and a small number of the members of his staff. The process is "brand new," and is based primarily on information and insights that the group had learned from consultants and in research focused on the software engineering process.

The custom process model for the pilot project is small, and has been quite stable since the project began, with only minor changes in the area of CM. No major difficulties were reported regarding the encoding of this process model in the process automation tool.

Outside of the development of the custom process model, the interviewee and his staff are clearly end users of the process automation system. They took no part in the development of the initial environment or the process automation tool. They have performed little additional tool integration.

Use of tools and technology

Prior to the receipt of the automated environment, the members of the pilot project used a stand-alone version of Cadre Teamwork for reverse engineering. The interviewee reported that Teamwork was also included in the tool suite of the integrated environment, but the version included was older than the stand-alone version and did not include some essential capabilities.

The pilot team was also using a stand-alone version of Microsoft Project for project management. The automated environment included Autoplan as a project management tool. The interviewee reported that moving from Microsoft Project to Autoplan was a step backward, even though the individual capabilities of Autoplan were greater. He attributed this situation to lack of good and complete integration of Autoplan to In Concert. It not clear whether the problem was due to limitations of the tools, or to a poorly thought out portion of the process

[2] These SEI personnel were not involved in the current study.

model (this particular portion of the process model was encoded by the environment integrator).

The interviewee also expressed reservations about the general integration of the environment. He stated: *We have not seen anything integrated in the suite.... Right now, I think if I put In Concert on a server, Autoplan on a server, Teamwork on a server, and DBStar on a server I would have the same situation as I have right now.*

The pilot project team is currently making little use of the metrics being collected by the automated environment, since these metrics are focused on code generation (which is beyond the scope of the pilot). However, the pilot team is collecting information on numbers and types of errors, and has written their own tool to help them collect and evaluate this data.

Overall, the pilot project is using only a small portion of the tools provided by the automated environment, along with an equally small portion of the process automation capability. Thus, no overall statements can be made about the quality of the process encoded. However, a number of problems have been reported with the process automation tool itself, including locking up and crashing. A number of these problems appear to be associated with bridges that move data from tool to tool and was written by the environment integrator. However, others are judged by the interviewee to be directly attributable to the process automation tool.

Team characteristics and experiences

The staff of the project is a mix of military personnel and civilian employees who are new to reverse engineering activities and software development. According to the interviewee: *None of us have ever done this, none of us are software engineers, none of us have ever developed code for applications.* However, the background experiences of the group included significant modeling experience, using structured analysis (SA), and IDEF techniques.

The (military) manager is noteworthy for his "can do" attitude, "hands on" management style, and his strong respect and support for his project staff. The interviewee indicated that his staff for the project was highly skilled, and not necessarily representative of the staff of the wider organization.

Transition and adoption

The pilot project is part of a high-profile effort to evaluate the automated environment capability. The interviewee provides frequent feedback to superiors, environment advocates, and environment contractors. The pilot itself is part of the transition strategy. Lessons learned from the pilot are intended to assist other organizations in adopting the environment.

Training is provided by subcontractors to the environment integrator. Early training has been problematic, with failures in the environment itself and poor preparation on the part of trainers. These problems have led to the replacement of the training subcontractor.

Initial training was offered in general areas such as functional analysis, software engineering, and system administration. The interviewee suggested that these training sessions included a great deal of general information that was not relevant to the use of the automated environment. While the specific information may have been embedded in this general information, it made little sense to the interviewee to pay for and spend time in an entire session to get a small amount of relevant information.

The interviewee also felt that there were problems with the diverse range of personnel being attracted to the training sessions. The range of skills and experience was too broad, resulting in less experienced trainees holding more experienced trainees back. The interviewee was instrumental in modifying the training policy to require prerequisite skills.

No specific training was initially offered on the use of the process automation tool. The interviewee felt that this was a significant oversight.

The reaction of the interviewee to the training problems was to stop sending people to the courses, and instead contract out for training by expert consultants and tool vendors. For example, Cadre was asked to train on the use of the Teamwork tool, and to help in designing a reverse engineering approach. The interviewee viewed this training as highly successful.

In addition to the training problems, the interviewee noted a number of administrative problems with the automated environment. These include

- The group installing the automated environment was ill prepared.
- License management within the automated environment was inflexible.
- Changes to hardware on which the tools run were complex.
- The users of the automated environment were not allowed to install system patches or updated versions of tools. Procedures for tool updates were unclear.
- It took approximately 7 person-months to administer the few tools being used.

One relative success in the transition activity was the Help Desk support offered by the prime contractor. The Help Desk was very responsive and answered many questions quickly, but the personnel did not know the answers to all questions, because they were not experts in the individual tools. Unfortunately, the end users were not allowed to call the tool vendors directly, since there was no maintenance/support contracts with the vendors.

Impacts and insights

The interviewee highlighted a number of additional insights. These include

- Data bridges between tools must be complete and must work for process integration to be of value. This suggests that process integration efforts may fail unless the larger problem of data integration between tools is addressed.
- End users must have direct relationships with tool vendors. A single interface with the environment integrator is not sufficient.
- For long-term success of process automation environments, efficient and timely approaches to upgrading component tools must be developed.
- End users of process automation environments must be voting members of a body that controls the direction of environment maintenance and enhancements.

C.13 Projects D and G, Interview 1

Business Characteristics

This interviewee served as a consultant to a large DoD-funded program, the objective of which was to investigate new and innovative ways to implement software. The program took more risks than a normal commercial enterprise would, since pursuit of this type of leading edge technology often fails. There were three demonstration projects involving three large government contractors. The consultant was a member of committees for technology transition. He started working with the three projects and ended up working with two (projects D and G in Tables 3-1 and 3-2). The major focus of his comments as was project D.

Process Improvement

In project D, a "big leap" approach was used to implement process improvement. While project teams were familiar with the SEI Capability Maturity Model, incremental improvement was rejected. The project wanted to have effective requirements management and project management, and it believed that domain engineering together with application engineering would achieve the necessary improvement. This was managed (sequentially) by several captains who were never able to implement domain engineering effectively. However, there were some good results with application engineering. In the estimation of the consultant, this was not surprising, as he did not believe that the theory behind domain engineering could be effectively put into practice.

Initially, the project did not know what its processes should be. Process definition was often done in the abstract, as they were unsure of what they were trying to accomplish. The consultant asked process definers "Who is the customer?" They answered, "we are" but, in his estimation, that was not right answer. Finally as their experience increased, they did "real

well." They had a 9 to 11 person team of process definers, who defined approximately six subprocesses. The development teams were the guinea pigs who acted as the customer. The developers would give requirements to process definers, and the definers would define the process that the developers would then try.

Application focus

Project D's focus was a combat control system that had many sub-systems. The interaction between these subsystems resulted in interface problems. The consultant indicated that the project came up with a new system architecture, and they wanted to demonstrate innovative ways to develop and maintain this system. Many subcontractors were involved in this project. The demonstration project involved about 30 people.

Tools and Technology

In the project, the process definition language IDEF0[3] was "going to solve all process definition problems." Previously, they had done much work defining as-is processes using IDEF0. They had pretty sophisticated, but manually implemented, processes.

The project put together a scheme in which four tools worked together. However, the consultant felt the need for two parallel efforts: an automated approach supported, for risk mitigation, by an equivalent manual approach. The consultant felt that this strategy would allow success to be declared even if the automated approach was not a full success. As it happened, the consultant was subsequently on a red team to analyze why tools were not working together. One tool that combined process definition and project management provided input to a Cleanroom tool. The plan was to use ProcessWeaver, but that tool was viewed as too rigid for process execution (see next paragraph). The integration of the process definition/project management tool and the Cleanroom tool to ProcessWeaver never worked successfully. Thus processes were enacted manually with manual invocation of the application tools.

The consultant felt that, with real processes, creative people do not respond to a process automation tool that says "you will start the activity now." He gave the example

Developers will rebel if the system states: you cannot start implementing until your low level design has been approved. It's better to define the artifacts that are to be produced and the goal states that these artifacts can be in. As a process definer, I can say: here are the exit criteria that I will impose on you under which you can officially declare that a goal state has been reached. As Project Manager I have every right to impose these criteria on you. I overstep my bounds if I tell you to use this method, or you will start this process at that point and not until you have finished something else. Typically a programmer has 20 things going

---

3. IDEF0 is very popular with DoD agencies.

on at once, none of which are finished. An engine that assumes things are serial or sequential will not work – it last about 2 days.

Another issue, relating the adapting a tool to the project's needs, was raised by the consultant. He stated

One major participant in Project D knew what process definition/enactment required, and the contractor felt obligated to use the process definition/project management tool being supported by this participant. Politically someone decided that Cleanroom was "good." The Cleanroom tool came with overlapping functionality with the project definition/project management tool. The project didn't like the "pure" Clean room process and suggested changes. However, the owners of Cleanroom refused to make changes as they did not consider this to be "Cleanroom." So, in effect, the Cleanroom owners said: "That's not the Cleanroom method so I won't talk to talk to you." They wouldn't change the tool to conform to the project's requirements.

Team characteristics and experiences

Process definition teams in Project D were headed by a major in the Air Force. There were four teams with three to five people in each. One captain picked up the concepts of process definition very well, and she drove that activity. She talked to people involved in the processes a lot, she had people review the work, and she made it happen. The consultant felt that it needed a strong person to take on that responsibility.

Transition and adoption

The end-user experiences in Project D were mixed. Two of the end-user teams were very receptive and welcomed process definers. These end users gave many constructive suggestions. Two captains worked on process definition and interacted closely with the developers of the automation software. Contractors were also end users of the these processes. However, because of poor interaction with the process definers there was less buy-in with these end users. The end users associated with the contractors believed an unfamiliar process would be imposed upon them.

The consultant believed that the group that started successfully using the automated process is the group that Project D is basing their success story on. The extent to which the process has been used is a little disappointing - it is not being used as widely as it could. To transition

this from a demonstration project into the wider community would involve a lot of other issues that they are not equipped to handle.

Impacts and Insights

The consultant indicated that he believed in starting on a pilot basis by defining a manually enactable process first. He would be very reluctant to use automated tools first. He then suggested the following:

An initial manual implementation lets you wring out a range of methodology issues and provides you with good appreciation of what a balanced approach to process definition and enactment is. That will arm you with the ability to impose a set of quite realistic requirements on the next tool developer/vendor who comes along and says "I can solve your process problems." If you talk to tool developers based on sound knowledge of what's really involved, you will be less inclined to accept at face value what the tool developer says.

You shouldn't have to swallow process automation all in one go. You can start with a database for metrics, defining artifacts and their states in the repository. This allows you to manage the artifacts, while letting the process drift by itself. People should own and be responsible for changing artifacts from this state to that state by that date. Later, you can add prescribed methods for doing these things, add process activities, link them together, develop exit criteria, form process networks, etc.

By keeping process definition divorced from management of artifacts, you get flexibility to throw out a process that's not working well and substitute a new process without perturbing the products or artifacts that you are working on. You can add several processes working from multiple viewpoints on the same artifact without perturbing the artifacts themselves. These things you learn from first enacting manually. End-users may say "you didn't prioritize any of my activities but I wish you would. I have 30 activities that I need help in prioritizing." However, don't tell them what they have to do or it will be rejected. These things allow you to gradually work your way up the automation scale.

Make sure at the executive level that the expectations are set right and limitations are understood. Many managers have silver bullet syndrome - they will listen to the first good story they hear from sales representative, and don't have anything else to base decision on, other than that the tool sounds good. Then it becomes law. This is how "politics" happens.

C.14 Projects D and G, Interview 2

Business Characteristics

As with the consultant in the previous interview, this interviewee served as a consultant to a large DoD-funded program, the objective of which was to investigate new and innovative ways to implement software. The program took more risks than a normal commercial enterprise would, since pursuit of this type of leading edge technology often fails. The consultant became

involved in process automation in the early 90's in order to understand what it really took to implement a process support system. He participated in two of three projects (Projects D and G in Tables 3-1 and 3-2) in the DoD program.

Process Improvement

The consultant originally had a plan for Projects D and G that was predicated on doing process improvement assessments. However, the idea was rejected by Project G because this project believed that the consultant would likely ask some pretty hard questions. The consultant felt that an assessment would have been extremely valuable to get a good mapping of what currently existed, and to identify areas of leverage. However he was told that would not to be permitted. As an alternative, he had discussions to identify the project's process characteristics.

Project G, which had little experience in defining processes, adopted a process for Cleanroom software development as their own. The consultant indicated that Project G was a CMM Level 1 organization, but had many smart people. They had produced software very successfully for years. However, like most software support organizations, they had frequent minor maintenance problems. Project D, also a Level 1 organization, was supported by a well-respected software contractor, and they produced good-quality code.

The consultant believed that if he could introduce 1) well tested project management processes, 2) effective inspection and CM processes, and 3) the Cleanroom methodology for Ada development, into a trained and committed CMM Level 1 organization, then the organization could be elevated to Level 2 or Level 3. On the other hand the consultant believed that if a Level 1 organization was not willing to adopt change, then technology and training will not help. However, the consultant did not think that any of the projects reached Level 2 as a result of the process automation initiative.

Application Focus

Project D developed a software environment to support a space command and control application. This application was a proof-of-concept to examine domain-specific architecture ideas. It was based on common sets of services for implementing surveillance applications, space being one. The space application was implemented using a common infrastructure for message processing, database support, and GUI services, etc. These were all needed to support other applications areas such as air and missile warning.

The software environment automated a Cleanroom engineering process, and was initially coded using the C language with supporting library services. Because of the hard-coding in the system, it was difficult to modify and resulted in long lead times when process modifications were introduced. The application was then more flexibly implemented using ProcessWeaver. Once experience was gained using Weaver, a project management front end

was developed to support task dispatching to the ProcessWeaver process support environment. The system has been field tested, and deployed.

The consultant suggested that a task leader has different needs than a project manager. He felt that a project manager needed abstractions at the level of PERT and Gantt charts. On the other hand, a task leader has technical tasks to delegate. He/she needed to see the portfolio of these tasks in a crisp view. What the consultant found (using ProcessWeaver) was that members of the project were recording processes in a notebook. Every time someone had an instance of the process, they would take one of these forms, copy it and put it in the book. So the process now became a book of forms. Thus the consultant believed that a huge paper-based system was created to support the enactment environment. The consultant said: *This is silly - let's automate it and use that as a front end with which to manage Weaver.* They were thus forced to do considerable thinking about what were the needs of dispatching. This resulted in the development of a tool to handle tasks above the ProcessWeaver level.

Tools and Technology

The consultant raised several technology issues as a result of his experiences.

Issue 1: On the need for a flexible tool with design capability, the consultant stated

We felt we needed a tool we could use during process design. We needed to design a process, have someone come in and take a look at it, see if they liked the way the tool interactions worked, given of course they recognized that automation was coming. We needed get things right from a user perspective and a flow perspective. This lead us to the use of Process-Weaver, the commercial process automation tool from Cap Gemini. With this we could do rapid prototyping more efficiently and of course, the idea is that when you have the last prototype, you have the process.

Issue 2: On the need to manage prototypes, the consultant stated:

We had the issue of having to manage prototypes as prototypes; you'd better make sure you do configuration manage. You must make sure that you can go back to yesterday's stuff if you want to get it back. That was a lesson learned that we forgot a couple of times.

Issue 3: On the use good software engineering principles, the consultant stated:

When you use rapid prototyping ideas in developing a process support application, make sure you don't forget your good software engineering principles. This kind of development can lead you to code-and-go, getting into a debugging mode, rather than thinking of each prototype from a functional verification point of view.

Issue 4: On the need for tool integration, the consultant stated:

Tool integration turned out to be a major challenge. After we learned how to deal with ProcessWeaver, we had to learn how to integrate that tool with a number of other tools.

Issue 5: On the need for globally accessible data, the consultant stated:

One of the things I liked about the "hard-wired" prototype we developed was that we could store process state and attributes in "data frames." Data frames could be looked at by anyone. So I had global state information. There was some practical benefit to that. Let's say I make a provision in the process program that a requirements change can come through. I have some people doing design work, and I know that the requirements change is going to affect the design work. I would then have the ability to look at this requirements change, and also take a look at which activities are running. I can then determine which activities would be affected if I make the change and send a stop work order on those activities. And I could do that automatically because I have access to the global process state data.

Issue 5: On the importance of artifact state, the consultant stated:

Artifact states are second often class citizens in automated process models. You need to have them being peer as far as their management support. That's what I like about the InConcert approach where I can do anything to the database schema that I want to. If I want to make artifact state a first class attribute I can do that and I can write process programs to poll these states. Software engineers are not going to work in nice sequential threads – they require massive parallelism, and the only way to achieve that, in my estimation is to have good control over artifact state.

Issue 6: On prototypes and software architecture, the consultant stated:

The one thing we recognized was - software architecture skeletons are good ideas. You need to interject some notions of functional description and functional verification for each of the prototype levels, so that you can say "given the fact that I'm going to make this my prototype goal, I'll actually do some algorithmic development on paper, reason about it, and make sure it's correct and then I can go deeper into the prototyping." In that way you can probably eliminate much of the code-and-go activities that our process model seemed to foster.

Issue 7: On modeling data state vs. control, the consultant stated:

Another consultant and I spent a lot of time convincing the automation team that state-change architecture was a good thing. However, they had the idea that they could handle process enactment by just learning about artifact states. There was no notion of any kind of process control – they believed that all process control could all be derived from artifact states – I believe an invalid assumption. I can conceive of very simple information systems that are

purely data driven, where you might be able to come up with a graph algebra that could prove that, but the moment there is any kind of management control, the model has to break down.

Team characteristics and experiences

The Cleanroom team in Project G consisted of four developers and a team leader, while the certification team had six personnel. There was some conflict, as the certification team wanted to do some things the development team were doing.

At Project D, the consultant provided process support on the specification process. The team doing the specification process had seven people. They were following a modified version of the Cleanroom process – using Booch's ideas. Even though they were using more paper than they had to, they felt that they were doing object oriented analysis, and that was important to them. The consultant felt that was not a problem so long as they still had a behavioral model of the software.

Transition and adoption

The consultant raised several issues related to the transition and adoption of process automation technology. These are paraphrased below.

Issue 1: On being familiar with the process

Project G had six months of manual Clean room experience, so the adoption of automation was not that difficult. What Project G saw in the process-centered environment was support for the Cleanroom process, a process that they understood. Project G had been using certain forms, and the tool closely supported these forms. They already knew the Cleanroom process. If you knew the Cleanroom process, you could use the tool. Everybody was focussed on the activity, and the tool became secondary to the goal. The tool is very important to the goal because of the support the tool provides, but it is not appropriate to introduce the tool first.

There were really no issues at all with adoption. Project personnel were learning something that they felt was benefitting them, and they were motivated by a strong team leader. He said "trust me, this will be good for you" and they believed him. This is not always going to happen, but this was a small, tight team, and it worked.

The project knew the process, the tool supported it, and the moment they saw it they thought it was very intuitive. End users were presented with a network of activities, and when you clicked on one, the activity would bring up a set of boxes and each box represented a milestone in the Cleanroom process. You would work through the activity, and when you finished, the boxes changed color – it was pretty nice. I was amazed that they were able to take to it so easily.

Issue 2: On being involved early in project definition

With the audience at Project D, we did an enormous amount of work trying to come up with the environment, trying to pre-think problems; when we got out there, the first thing they said

was "The project is already underway, what do you have that will help us. We are not really interested in this process stuff, we look at that as a diversion to what we want to do." They wanted to have nothing to do with Cleanroom, but we had already done a lot of work in this area as one of our key technologies. They said that they wanted to adopt the Ada process model. On the fly we were doing an enormous amount of work comparing Cleanroom vs. the Ada process model and saying "where do these match, where do they differ, and how are we going to handle this?" So I found that I had to do new processes. I could no longer count on legacy processes I thought I could count on. So a lot of work had to be done.

Issue 3: On "big bang" vs. incremental approaches.

Project D personnel wanted to know the details of the technology right away; they wanted to understand what they were in for. However, I said I wanted to introduce it incrementally. When you rapidly introduce a lot of new technology concepts, you get a lot of blank stares, then you get apprehension, then you may find your audience turns downright angry. When you have to go back and say "Let's just deal with this a piece at a time", you may get again, "Oh no I want to know it all!" We overcame this problem by letting different individuals become experienced with different parts of the technology. So the person who learned tool A is not the person who learned Tool B. In that way we introduced some separation of concerns, and got cross-fertilization. If we had lost one person we did not loose all the expertise. In addition, the technology has got to be supportive of human activity, it's got to be very goal oriented, and produce an immediate result.

Issue 4: On vested interests in process notations

Because of a former project manager in Project D, every person in the management chain used IDEF0. There was therefore a legacy that one represented business processes in IDEF0, – personnel already had it institutionalized. If we had known that in advance, then we could have planned for this fact. When it came time to define an enactable process that I could hand off to an engineer, IDEF0 was not a good representation for that.

People at Project D spent enormous amounts of time trying to come up with IDEF0 process diagrams that people could agree to. Typically, one subcontractor took a look at these diagrams and said "why can't you do functional flow or data flow and make it as clear as possible. Don't try to do all this obfuscation" As devil's advocate, I had to say to Project D: "this is great work, you are learning a lot of things, and recording a lot of issues". We (as consultants) were also introducing ideas that were conflicting with their own. It did not make for a harmonious situation, and took almost a year for us to feel part of the organization.

Issue 5: On consulting tactics

In my opinion, a good consultant will never come in and say "everything you are doing is bad". You can't say that, so what you have to do is to back into these things. I have two experiences here, one was a honeymoon (Project G) and the other was hell (Project D). The lesson I learned was that the only way you can reach people is through education. I have to empirically

write issues out and make them so obvious that you ignore them at your peril. If I haven't done that then I have not done my job as a consultant. If somebody says they want to use IDEF0, and I see that there is some value there, then what I would say is "I want you to keep IDEF0, because you are doing a good thing, but I want you to recognize that it's not enough," and you have to show them why it's not enough.

Issue 6: Supporting champions

You have to find out where the best injection point is. You've got to give that person who is the champion a success. If you come in and give him a failure you are done for.

Impacts and Insights

The consultant felt that, when Project G worked on the Cleanroom tasks, they had a good intellectual basis to support them, and they knew what their objectives were. When they went back to "design as usual" they found that development was more haphazard. The consultant stated that, a typical project remark might be: "Cleanroom: well defined specs, management seems to like this approach, we are getting good results, we produce behavioral specs on the software objects before we think of implementing it, we are getting better intellectual control over our problems and our customers like it a lot better too." The consultant believed that Project G got "religion" by successfully bidding on two contracts. They also received positive reinforcement from management. In the eyes of the consultant, a typical management remark might have been: "These specs look great! It may have taken three times as long but this is the best spec I ever saw." This made the developers feel highly motivated. When the automated tools came along, they got additional support in following this process and they were in an open frame of mind to accept the computer support for Cleanroom.

In comparison, Project D had several significant issues that had to be resolved. First, the consultant was not involved early in the project and was not able to influence the early direction of the effort. Because of this, the existing team members resented the consultant when he suggested different approaches to issues. The consultant had a significant success with Cleanroom methodology (with Project G) and was strongly motivated to see its success broadened. However the focus at Project D was on the Ada process model, not Cleanroom. The consultant did not consider the technique used to define processes at Project D (i.e., IDEF0) adequate to provide the degree of rigor needed for process automation. However, IDEF0 was quite entrenched, so compromises had to be made.

References

[ISO-91] *Quality Management and Quality Assurance Standards - Part 3: Guidelines for the Application of ISO-9001 to the Development, Supply and Maintenance of Software.* Geneve, Switzerland: International Organization for Standardization, 1991.

[Paulk-93] Paulk, M. C., et al. *The Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24, ADA 263403). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

Acknowledgments

This report could not have been possible without the support we received from the many individuals that we interviewed. We appreciate the time they spent on our behalf, and how direct and honest these individuals were with us. Because we guaranteed their anonymity, we cannot name the interviewees. However, you know who you are! Thank you.

The following trademarks and servicemarks are used in the report:

AIX is a registered trademark of International Business Machines Corporation
AutoPlan is a registered trademark of Digital Tools, Inc.
CCC is a registered trademark of Platinum Technology Inc.
Capability Maturity Model is a registered service mark of Carnegie Mellon University
CMM is a registered service mark of Carnegie Mellon University
Data Access Element is a registered trademark of Neuron Data Inc.
DBStar has been applied for as trademark of DBStar, Inc.
DDTs is a registered trademark of QualTrack Corporation
Interleaf is a registered trademark of Interleaf, Inc.
FlowMark is a trademark of International Business Machines Coproration
FrameMaker is a registered trademark of Frame Technology Corporation
HP is a registered trademark of Hewlett-Packard Company
IMS is a registered trademark of International Business Machines Coproration
Macintosh is a registered trademark of Apple Computer, Inc.
Lotus Notes is a registered trademark of Lotus Development Corporation
Oracle is a registered trademark of Oracle Corporation
Pentium is a registered trademark of Intel Corporation
ProcessWeaver is a registered trademark of Cap Gemini Sogeti
Refine/C is a registered trademark of Reasoning Systems Inc.
Schedule Publisher is a registered trademark of Advanced Management Solutions
Softbench is a registered trademark of Hewlett Packard Company
Software Through Pictures is a registered trademark of IDE, Inc.
Sun is a registered trademark of Sun Microsystems, Inc.
Synervision is a registered trademark of Hewlett-Packard Inc.
TeamWork is a registered trademark of Cadre Technology Inc.
Unix is a registered trademark exclusively licensed through X/Open Company, Ltd.
VAX/VMS is a registered trademark of Digital Equipment Corporation
WordPerfect is a registered trademark of the Corel Corporation
WorldView is a registered trademark of Interleaf Inc.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-96-TR-013	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESC-TR-96-013
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-95-C-0003
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO 63756E; PROJECT NO. N/A; TASK NO N/A; WORK UNIT NO. N/A
11. TITLE (Include Security Classification) Software Process Automation: Experiences from the Trenches	
12. PERSONAL AUTHOR(S) A. Christie, L. Levine, E. Morris, D. Zubrow, T. Belton, L. Proctor, D. Cordelle, J. Ferotin, J. Solvay	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM TO	14. DATE OF REPORT (year, month, day) July 1, 1996 15. PAGE COUNT 94
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
FIELD	GROUP	SUB. GR.	process-centered environments, software environments, software tool integration, technology adoption, adoption guidelines, technology inhibitors, interviews, experiential report, lessons learned

19. ABSTRACT (continue on reverse if necessary and identify by block number)

Software process automation is a new technology with significant promise. However practical experience in the field is still limited and there appears to be a variety of potential barriers to its use. The objective of this empirical study is to document current practical experience and to identify what works and what does not. Lessons learned from the study will be disseminated to help others who wish to implement the technology. This report documents results from the first phase of the study in which 14 in-depth interviews were conducted. Personnel interviewed were involved in projects in which process-centered environments were developed and adopted.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631 22c. OFFICE SYMBOL ESC/ENS (SEI)